An Integrated Cloud-Based Framework for Remote Sensing Data Processing

Ghaffar, M. A. A.,[1] Vu, T. T.[1] and Maul, T.[2]
[1]School of Geography, The University of Nottingham, Semenyih 43500, Malaysia
[2]School of Computer Science, The University of Nottingham, Semenyih 43500, Malaysia

Abstract
Nowadays, an advanced remote sensing technology acquires huge amounts of Earth surface details with multi-spatial, multi-temporal and multi-spectral resolutions that have changed drastically the size and structure of data. In order to process such big remote sensing data, it is vital to adopt a new approach, i.e. Cloud Computing, which is an elastic, scalable and reliable solution. In line with current deployment of data storage on the cloud by various space agencies and data providers, we are developing remote sensing processing services on the cloud to take advantage of cloud-based datasets and provide remote sensing communities with a more suitable approach to deal with today's remote sensing big data. This paper describes how the proposed framework components interact with each other and describes a proof of concept of supervised classification software as a service (SAAS) built on top of Amazon Web Services public cloud working on Landsat 8 data sets. Other more complicated image understanding services will be implemented in further studies.

1. Introduction

Recently obtained Earth Observation (EO) data have prompted remote sensing scientists to develop more complicated and novel image processing algorithms (Braaten et al., 2015). Given more advanced types of sensors, satellites can transmit a more detailed description of the Earth captured at higher spatial, spectral and temporal resolutions. As a result, remote sensing datasets have become larger in size and more complex with reference to structure (Padarian et al., 2015 and Plaza et al., 2011). For example, Landsat5 uncompressed scene size is approximately 366MB while Landsat8 scene size is more than 5 times greater (See Table 1). From a structural perspective, Landsat 8 scenes have 16-bit pixel values with 11 bands while Landsat TM and ETM+ have 8-bit pixel values with 6 and 7 bands respectively (USGS 2013) (NASA 2015). Despite facing speed and performance difficulties, desktop workstations were previously capable of managing EO data. However, desktop workstations have rapidly become computationally insufficient as a result of the EO data evolution. In order to process large datasets and derive information from them in a timely manner, computing power, input/output operations, data management, filesystem and data storage factors have to be re-considered (Ma et al. 2015). Remote sensing scientists have had to adopt new techniques and methods to store, analyse and process such big data. Over the last decade, Grid and High Performance Computing (HPC) usage has grown rapidly. Research studies have been conducted towards utilizing HPC to afford the required computing power for EO data processing tasks (Liu et al., 2015). Although HPC has solved the performance issue, scalability is still considered a challenge (Cavallaro et al., 2015). The aforementioned remote sensing big data raise the demand for more elastic and reliable solutions. More recently, different Cloud computing layers such as SAAS Software as a Service, PAAS Platform as a Service and IAAS Infrastructure as a Service are proposed as solutions to accelerate computing tasks through elasticity and scalability enabled options. (see Figure 1). The SAAS model is considered as a software distribution layer to host and publish the application to end users. PAAS is a set of tools designed to ease the development and deployment of these applications efficiently, while IAAS is the pool of hardware resources which are used for virtualization (Zhu et al., 2014). Using the aforementioned topology in Figure 1, cloud computing can offer new scalable and cost effective solutions to process remote sensing big data. This paper presents a proof of concept of our development towards a full set of processing services on the cloud. A land cover supervised classification service is developed on top of Amazon Web Services (AWS) using a simple scenario and ready built libraries to process EO big data hosted on AWS Simple Storage Service (S3) following user demands. Section II describes how the system works and how different cloud

computing models can communicate with each other to deliver the output. Section III discusses the results and the system interface. Finally, Section IV describes relevant works to be done in further studies.

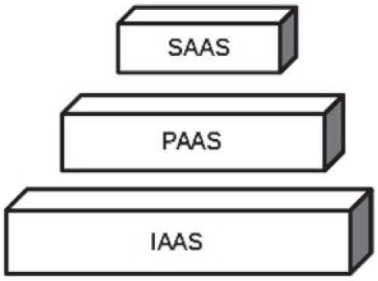

Figure 1: Cloud Computing Architecture

Table 1: Differences in Landsat dataset size

Data set	Landsat5	Landsat7	Landsat8
Compressed	160MB	900MB	1GB
Uncompressed	366MB	1.6GB	2GB

2. Methodology

2.1 Cloud Computing Superiority

Cloud computing represents a state of the art and trending methodology in big data analysis in general and EO data in particular (Zinno et al., 2015 ;Ghaffar and Vu, 2015 and Bica et al., 2014). Earth Observation agencies and satellite image providers usually have their own private cloud infrastructure to store and process data. However, there was always the question of how to minimize data transfer cost, especially when dealing with big data. Hence, well-known agencies like NASA or ESA tended to have agreements to share EO data through public cloud services like AWS or Microsoft Azure (MODISAzure, 2015) (NASA NEX, 2015), so there has been tendency for data to become bigger and open for all research and scientific use. Such agreements solved one issue of big data which was data storage and transfer cost, leaving processing issues to the creativity of the community and according to the needs of different researchers. The proposed framework used one sample of the recently opened EO data set, Landsat 8.

2.2 Earth Observation Data

Taking advantage of the elasticity options, space agencies like NASA and ESA store their EO data on different cloud storage services (NASA NEX, 2015 and Landsat on AWS, 2015). In early 2015, in support of the White House's Climate Data Initiative (Climate - Data.gov, 2015), Amazon committed to make Landsat 8 imagery data widely available through AWS public datasets. AWS is a collection of web services used to manage remote computing resources on the Amazon cloud platform. The proposed system aimed to utilize the aforementioned EO data which became accessible for free on the cloud. As a public dataset, Landsat 8 scenes are available via Amazon S3 Simple Storage Service within hours of production. Amazon S3 is a scalable and reliable object storage service which allows storing and archiving of such big data like EO data. Not only can users access the recently obtained scenes but they can also use datasets archive since 2013. AWS provides a list of all the available scenes on the Landsat 8 bucket with some meta data including acquisition date, cloud coverage and scene id. These Meta data are harnessed to set the download criteria according to the end user choice whereas the system takes advantage of the ready hosted Landsat 8 datasets on Amazon S3 storage service.

2.3 Supervised Classification

Relying on the experience of the image specialist, supervised classification techniques allow the creation of training samples which become the basis of setting up statistical parameters used in classification or regression of data outside the training set (Cavallaro et al., 2015). For example, Support Vector Machines (SVMs) (Cortes and Vapnik, 1995), are one of the most robust supervised classification techniques being used nowadays. A core SVM idea lies in segregating data instances which belong to disparate classes by identifying the maximum margin hyperplanes that separate them. SVMs are attuned to the high-dimensional essence of remote sensing data like multi-spectral bands satellite images. They include three main steps:

1. To create the training dataset where the user selects representative samples for each land cover, which are later used to identify the land cover classes in the entire image.
2. To train the SVM model whereby the algorithm takes the training data set (including the label/target data) as an input to produce a model that captures the underlying structure of the data samples. This model will be used to predict land cover classes among new data sets (non-training data samples).
3. Take the trained model as an input to classify new data (entire image) according to the underlying structures that have been captured by the model.

SVM learning is a commonly used technique in remote sensing data applications e.g. forest mapping, land cover and land use monitoring, urban development and risk management. Hence, we have

chosen to deploy it over the Amazon Cloud Computing platform utilizing the available scalability options.

2.4 System Architecture and Workflow

The framework is fully developed and running on top of AWS cloud infrastructure. As a client-server architecture, the main feature of the framework is that the user doesn't need to purchase an expensive and advanced PC or upgrade hardware in order to execute image classification tasks on the scale of big data. Instead, users can just submit a request through a thin client or a very limited workstation connected to the Internet to the cloud-based server and it will execute all the required processes through a processing chain.

From Figure 2, we can see that the system has a workflow where we can identify 3 main steps as follows: 1) Data acquisition, 2) Data processing chain, and 3) System output and final results.

2.4.1 Data acquisition

The conceptual framework offers two options to set the Area of Interest AOI through the front-end interface which asks the user to either draw a polygon through Openlayers (OpenLayers 3, 2015) Web Map Service (WMS) base layer or directly upload a polygon shapefile representing the AOI. As illustrated in Figure 3 and after uploading or drawing the AOI, the system asks the user to define the scene's selection criteria, i.e. scene acquisition date range and percentage of cloud coverage.

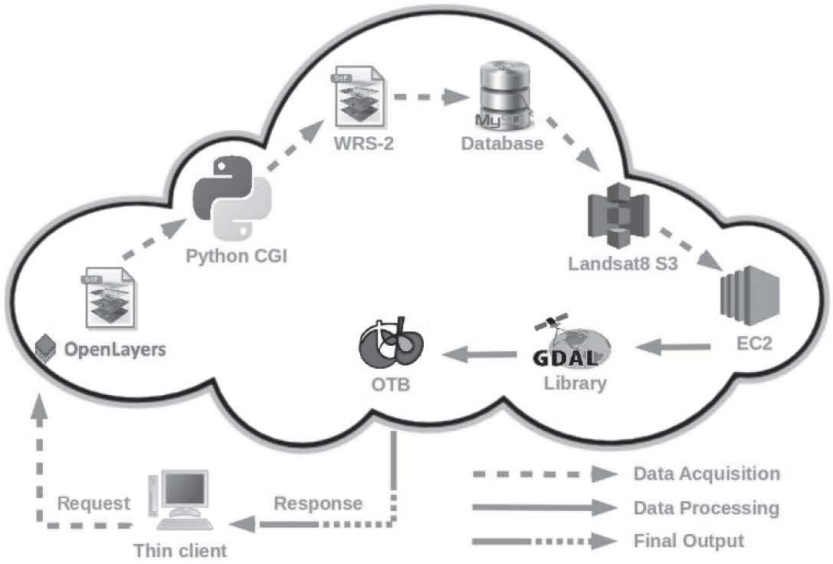

Figure 2: System Workflow

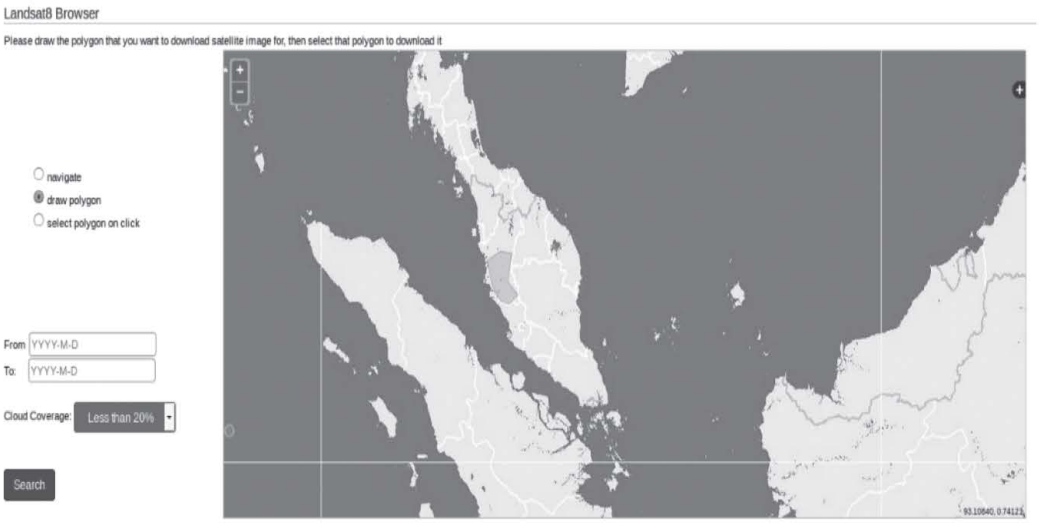

Figure 3: System Interface

Once the user has submitted the request loaded with the selection criteria, from the server side, a Python CGI (Python Cgi, 2015) script runs to check if there is an intersection between scenes to form the drawn or uploaded polygon. In order to achieve this task, WRS-2 Worldwide Reference System shapefile has been used alongside with Fiona library (Fiona, 2015) to detect the scenes which need to be merged. Using the aforementioned Python script, the system can retrieve all the scenes which are matching with predefined selection criteria. Amazon provides a database which contains all the Meta data of the stored scenes in Landsat 8 public datasets. Using the scene's URLs which are retrieved from the database, the next step is to directly access the Amazon S3 bucket through the URL: http://landsat-pds.s3.amazonaws.com/ and download the required scenes to Amazon EC2 Elastic Compute Cloud instance. EC2, the IAAS offering from AWS, offers a wide range of instance types to fit in different kinds of applications. Considering that image processing is a compute-intensive task (Zhang et al., 2010), the system relies on compute optimized instances.

2.4.2 Data processing chain

The Geospatial Data Abstraction Library (GDAL) (GDAL, 2015) is utilized by the system to merge all intersected scenes and build the mosaic. GDAL is also used to crop the resulting mosaic according to the drawn or uploaded polygon of the AOI. Once the AOI is ready to start the classification processing chain, a set of system calls are used to employ the Orfeotoolbox (OTB) library. Based on the OpenCV machine learning framework, OTB extends the functionalities of the libSVM library to apply supervised classification tasks using different algorithms such as SVM, Boosting, KNN, Random Forests and Normal Bayes. However, the SVM algorithm was chosen here as a proof of concept, but any other classifier could be used. The first system call sent from the web interface is for otbcli_TrainSVMImages-Classifier, which needs the AOI raster image, a shapefile containing the labelled data as input, and generates the training model as output. The interface provides two options, either to draw the training vector data in the form of polygons which it is automatically stored as a shapefile, or to upload it as a shapefile directly through the upload file button. The second system call is for otbcli_ImageClassifier, which takes the new data set and the training model as input and generates the new labeled data as output. The third system call is for otbcli_ColorMapping, which takes the new labelled data as input along with a predefined colouring theme to replace the categorized classes with their matching colors.

2.4.3 System output and final results

Since the expected output consists of processed images, one of the most popular ways to share and publish geospatial data is the use of GeoServer. GeoServer is designed to serve any major spatial data source whether it is raster or vector data sets. End-user has two options to visualise the output, either to download the processed data sets locally, or to host the output data on a cloud-based GeoServer instance (see Figure 5). Hosting the output on GeoServer allows using it from different applications and client-side libraries e.g. OpenLayers and Leaflet.

3. Results and Discussion

The proposed system is implemented in Python along with an extended set of open-source geospatial libraries. As a platform hosting all data processing and storage tasks, Amazon EC2 and S3 services are used to leverage cloud computing from scalability and cost efficiency perspectives. A simple experimental setup was implemented to test both scalability options and cost efficiency. Amazon Elastic Load Balancer (ELB) is used to test how the system can proceed against the expected workload. Auto scaling is used to ensure that the proper number of instances is running to serve the web application without additional fees. ELB allows the application developer to set a minimum and maximum number of instances to run at the backend. It ensures that the number of instances will never go above or below the maximum or minimum number of instances automatically. Moreover, it keeps checking the status of the auto scaling instances group and in case there is a failure in one of the instances, it redistributes the workload to the rest of the instances until the out-of-service instance is recovered or replaced with a new one. Table 2 shows a simple comparison between the time consumed to classify two merged scenes of Landsat8 datasets through a standalone C4.large instance (2 CPU, 4GB memory approximately) and a group of 4 instances using the auto scale and ELB service. ELB was set to work once the utilization of the CPU reaches 80%, with minimum of 2 instances and a maximum of 4. Table 3 shows the running cost of both cases when a Pay-As-Use model is chosen. AWS charges clients per hour, which means that even if the instance worked for a few minutes, it is still considered one hour.

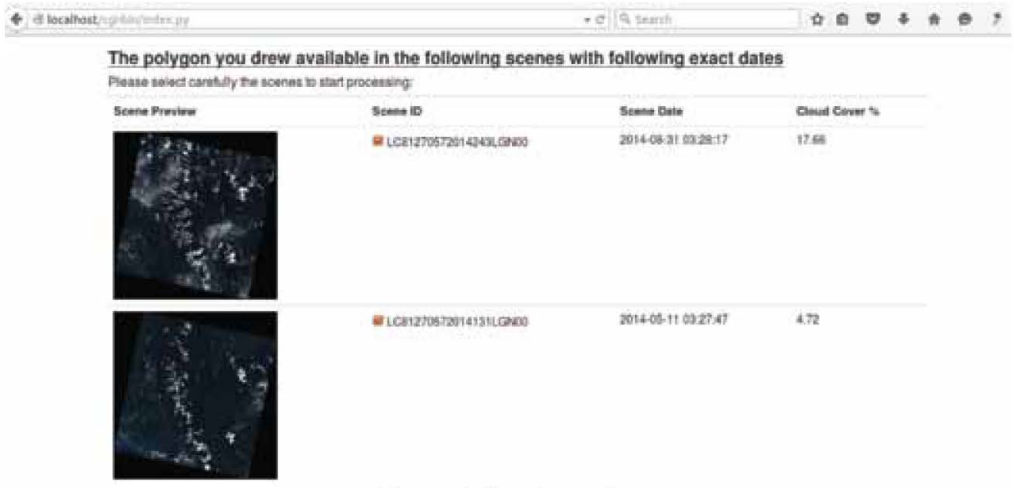

Figure 4: Search results

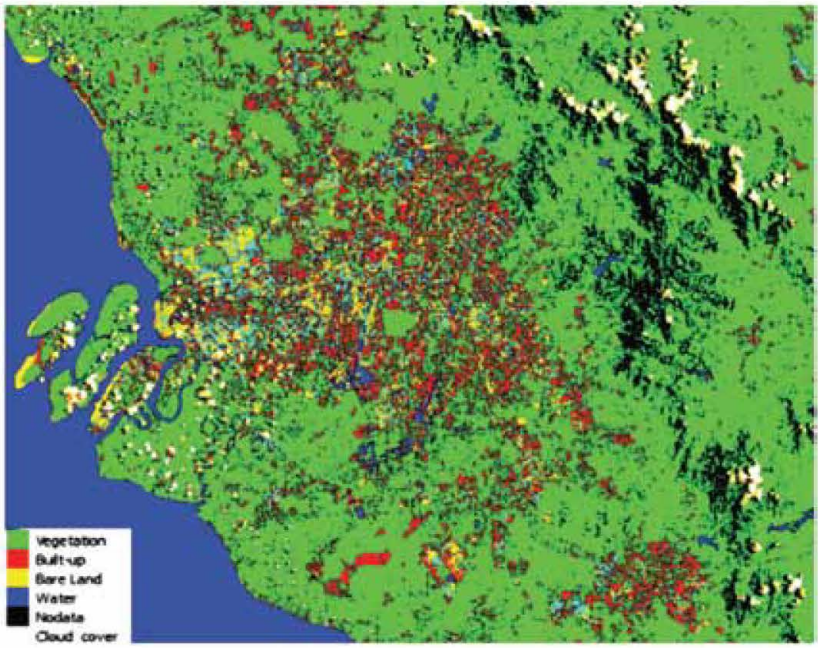

Figure 5: Classification map

Table 2: Average of processing time per 10 requests of a C4.Large instance

Instances #	Average of time in seconds
1	35
4	17

Table 3: Running cost per 10 requests on C4.Large instances in USD

Instances #	Cost per 1 hour
1	.105
4	.42

Hence, the cost of running 4 instances is much higher than running a standalone instance. From Tables 2 and 3, Cloud computing models show how the performance could be resilient, scalable and robust according to different system requirements. Moreover, the running cost is very low compared to on-premises infrastructure and hardware upgrades. The main advantage of using AWS scalability options is to take full control of how an application can react to peak usage time and leverage the EC2 machine either by scaling up the computing power (represented in the number of VCPUs) or scaling it down during non-active or low usage time.

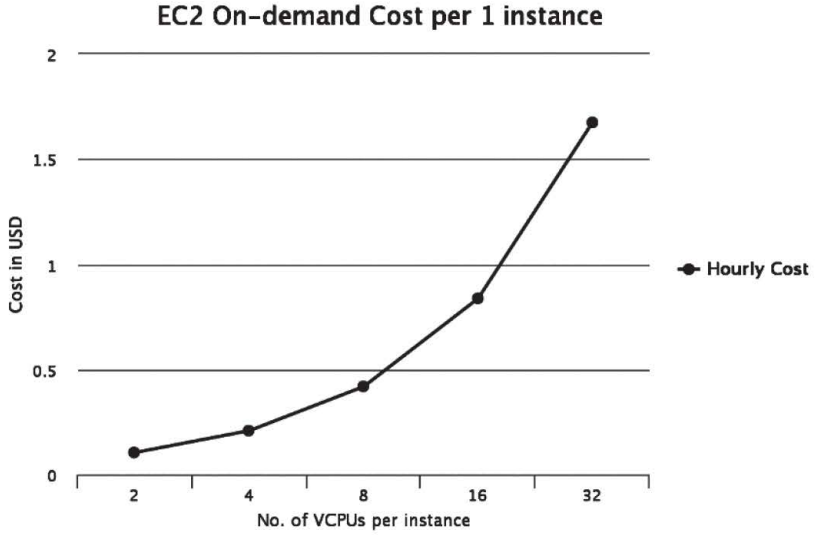

Figure 6: EC2 cost compared with number of VCPUs

Figure 6 shows a comparison between cost and the effect of increasing the number of VCPUs on the On-demand hour rate per EC2 instance. It is noteworthy that the relation is directly proportional once the number of VCPUs is doubled. However, the cost of using different number of EC2 instances couldn't be evaluated because the minimum unit of EC2 usage is 1 hour even if the process took few minutes. Future experiments with much larger data sets and more complicated processing, which shall take longer than 1 hour, will enable such a comparison.

4. Conclusion

This paper proposed an integrated framework capable of hosting processing chains to perform satellite image processing tasks utilizing ready built libraries and analytical tools. The framework mainly focused on associating free data hosted on cloud storage servers i.e. Landsat 8 data sets with ready-built open-source processing tools to generate software as a service application. The experimental system is considered as an initial development which opens doors to more complex processing services that make use of a wide variety of cloud computing models and services.

5. Future Work

Further development will involve more complex image processing and understanding algorithms, and applying different neural network and machine learning techniques for feature extraction services. Novel algorithms derived from extensive experimentation need to be applied to newly developed services and to be investigated in terms of user feedback. A comparison study can be carried out to show the differences I n performance and cost efficiency between cloud computing as a platform or standalone desktop computing in addition to including other factors like input/output operations, filesystems, and storage costs.

Acknowledgments

This study is part of a project funded by FRGS Malaysia Ministry of Education, grant no. FRGS/2/2013ICT07/UNIM/02/1. The work is also supported by AWS Education Research Grant.

References

Bica, M., Bacu, V., Mihon, D. and Gorgan, D., 2014, Architectural Solution for Virtualized Processing of Big Earth Data. *Proceedings - 2014 IEEE 10th International Conference on Intelligent Computer Communication and Processing*, 10.1109, 399-404.

Braaten, J.D., Cohen, W.B. and Yang, Z., 2015, Automated Cloud and Cloud Shadow Identification in Landsat MSS Imagery for Temperate Ecosystems. *Remote Sensing of Environment*, 169, 128-38.

Cavallaro, G., Riedel, M., Richerzhagen, M., Benediktsson, J.A. and Plaza, A., 2015, On Understanding Big Data Impacts in Remotely Sensed Image Classification using Support Vector Machine Methods, *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 99, 1–13.

Climate - Data.gov, 2015, Data Related to Climate Change, [Online], Available: https://www.data.gov/climate/.

Cortes, C. and Vapnik, V., 1995, Support-Vector Networks. *Machine Learning*, 20, 273-97.

Fiona, 2015, Fiona Package Index, [Online], Available: https://pypi.python.org/pypi/Fiona.

GDAL, 2015, Geospatial Data Abstraction Library, [Online], Available: http://www.gdal.org/.

Ghaffar, M.A.A. and Vu, T.T., 2015, CloudComputing Providers for Satellite Image Processing Service: A Comparative Study, *International Conference on Space Science and Communication (IconSpace)*, 61–64.

Liu, J., Feld, D., Xue, Y., Garcke, J. and Soddemann, T., 2015. Multicore Processors and Graphics Processing Unit Accelerators for Parallel Retrieval of Aerosol Optical Depth From Satellite Data: Implementation, Performance, and Energy Efficiency. *Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal of, 8(5), pp.2306-2317.*

Landsat on AWS, 2015, Landsat On AWS [Online], Available: http://aws.amazon.com/public-data-sets/landsat/

Ma, Y., Wu, H., Wang, L., Huang, B., Ranjan, R., Zomaya, A. and Jie, W., 2015, Remote Sensing Big Data Computing: Challenges and Opportunities, *Future Generation Computer Systems*, 51, 47–60.

MODISAzure, 2015, Accelerating the Pace of Environmental Research, [Online], Available: http://research.microsoft.com/en-us/projects/modisazure/.

NASA NEX, 2015, [Online]. Available: http://aws.amazon.com/nasa/nex/.

OpenLayers 3, 2015, A High-Performance, Feature-Packed Library for all your Mapping Needs.[Online] Avaiable: http://openlayers.org/.

Padarian, J., Minasny, B. and McBratney, A.B., 2015, Computers and Geosciences using Google'S Cloud-Based Platform for Digital Soil Mapping, *Computers and Geosciences*. 83,80-88.

Plaza, A., Du, Q., Chang, Y.L. and King, R.L., 2011, High Performance Computing for Hyperspectral Remote Sensing, *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*. 4,528-544.

Python Cgi, 2015, [Online], Available: https://docs.python.org/2/library/cgi.html.

USGS, 2013, What Are the Band Designations forthe Landsat Satellites?, [Online]. Available: ,http://landsat.usgs.gov/band_designations_landsat_satellites.php.

Zhang, N., Chen, Y.S. and Wang, J.L., 2010, Image Parallel Processing Based on GPU, *Proceedings - 2nd IEEE International Conference on Advanced Computer Control ICACC*,3 ,367-370.

Zhu, H., Zhang, C. and Ye, J., 2014, Research on Remote Sensing Network Control using Cloud Computing Services, *Proceedings of 2014 IEEE International Conference on Mechatronics and Automation*, 363-367.

Zinno, I., Mossucca, L., Elefante, S., De Luca, C., Casola, V., Terzo, O., Casu, F. and Lanari, R., 2015, Cloud Computing for Earth Surface Deformation Analysis via Spaceborne Radar Imaging: A Case Study, *IEEE Transactions on Cloud Computing*. 7161, 1–35.