Implementing Scalable Geoweb Applications Using Cloud and Internet Computing

By
Seyed Emad Mousavi
BSc Geomatics Engineering

A Thesis Submitted in Partial Fulfillment
of the Requirements for the Degree of

Masters of Science in Engineering

In the Graduate Academic Unit of Geodesy and Geomatics Engineering

Supervisors: Yun Zhang, PhD., Geodesy and Geomatics Eng.

Monica Wachowicz, PhD., Geodesy and Geomatics Eng.

Examining Committee: David J. Coleman, PhD., Geodesy and Geomatics Eng.

Raid Al-Tahir, PhD., Geodesy and Geomatics Eng.

Weichang Du, PhD., Faculty of Computer Science.

UNIVERSITY OF NEW BRUSWICK
March 2014

©Seyed Emad Mousavi, 2014

ABSTRACT

New advancements in technology such as the rise of social networks have led to more geospatial data being produced every day. The current issue with the large volume of geospatial data is to store and process it because of the scalability of the data. In this thesis, two computing implementations, cloud computing and Internet computing, are studied and evaluated for their capability in storing, processing and visualizing large volumes of geospatial data. For the cloud computing implementation, the different concepts of cloud computing have been analysed according to their applications, models and services. Moreover, a case study using cloud computing platforms has also been implemented for storing and processing geotagged tweets retrieved for a national recreational park in Vancouver, BC. For the Internet computing platform, the Open Geospatial Consortium's Web Processing Service has been investigated as a framework for sharing geospatial data and processing it over Internet. A raster calculation algorithm in Web Processing Service platforms has also been implemented on 2 scenes of Landsat satellite imagery to evaluate WPS' capabilities in handling large volume of data. Results of this research suggest that internet computing can be used to handle geospatial data processing but, when dealing with large volumes of data, this study proves that Internet computing and current Geospatial Information Systems are not suitable to be used and cloud computing platform can be utilized to handle large volumes of geospatial data.

ACKNOWLEDGEMENTS

It is a pleasure to acknowledge and thank the people whom without this thesis would not be possible:

I am highly grateful to my supervisor Dr.Yun Zhang for his important support throughout this work. His understanding, encouraging and personal guidance have provided a good basis for the present thesis. It is difficult to overstate my gratitude to my supervisor Dr.Monica Wachowicz, with her enthusiasm and her inspiration throughout my research period, she provided encouragement, sound advice, good teaching, great feedback, and lots of good ideas. I would have been lost without her.

David Fraser, Sylvia Whitaker, Lorry Hunt and Amir Abouhamze who were always available when needed and have always kindly offered their assistance.

I am grateful for the open source tools and projects used in this research.

I am deeply indebted to my parents, without whom I would not be standing where I am, they taught me to pursue my dreams, they taught me not to give up, they taught me to live, they taught me to love, they are the greatest teachers of my life.

And last but not the least; I would like to thank my lovely wife, Samira. She lost a lot due to my thesis but she always gave me hope when there were none, without her understanding and encouragement, this thesis would not be possible.

DEDICATION

To:

Samira

With Love

TABLE OF CONTENTS

ABSTRACT ii

ACKNOWLEDGEMENTS iii

DEDICATION iv

TABLE OF CONTENTS v

LIST OF TABLES x

LIST OF FIGURES xi

LIST OF SYMBOLS AND ABBREVIATIONS xiii

Chapter 1 1

1 Introduction 1

1.1 Research Background 2

1.2 Research Challenges 4

1.3 Research Objectives 6

1.4 Methodology 7

1.5 Overview of the Chapters 8

Chapter 2 14

2 A Comparative Study of Web Processing Service Implementations for Raster Calculation 14

Abstract 14

2.1 Introduction 15

2.2 Definition of WPS 16

2.2.1 Processes of WPS 18

2.2.2 WPS Operations 18

2.2.3 Data Types in WPS 20

2.2.4 Advantages of WPS 20

2.3 WPS Implementations 24

2.3.1 52°North Project 24

2.3.1.1 Features of 52°North 24

2.3.1.2 52°North Architecture 25

2.3.1.3 Raster Calculations in 52°North 27

2.3.2 ZOO Project 27

2.3.2.1 Raster Calculations in ZOO 29

2.3.3 Deegree Project 29

2.3.3.1 Deegree Architecture 30

2.3.3.2 Raster Calculation in Deegree 31

2.3.4 PyWPS Project 32

2.3.4.1 PyWPS Architecture 32

2.3.4.2 PyWPS Raster Calculations 33

2.3.5 Geoserver 33

2.3.5.1 WPS Processes in Geoserver 34

2.3.5.2 Process Chaining in Geoserver 35

2.3.5.3 Raster Data in GeoServer 36

2.3.5.4 Raster Calculation in Geoserver 37

2.4 Comparative Study 39

2.5 Conclusions 43

Chapter 3 48

3 A Comprehensive Overview of Cloud Computing in GIS 48

3.1 Introduction 49

3.2 Definition of Cloud Computing 51

3.2.1 Cloud Computing as a New Computing Paradigm 55

3.2.2 Cloud Computing as a New Framework 56

3.2.3 Cloud Computing as a New Computing Architecture 57

3.2.4 Cloud Computing as Services 59

3.2.5 Cloud Computing as a Deployment Model 60

3.3 Storing Data in Cloud Computing 63

3.3.1 Manage and Process Data in Cloud Computing 64

3.3.2 Visualizing Data in Cloud Computing 66

3.4 Cloud Computing Advantages and Disadvantages 68

3.4.1 Advantages of Cloud Computing 68

3.4.2 Disadvantages of Cloud Computing 70

3.5 Applications of Cloud Computing in GIS 71

3.6 Conclusions 72

Chapter 4 78

4 Designing a Scalable Cloud Implementation for Mapping Geotagged Tweets 78

4.1 Introduction 79

4.2 Cloud Computing in GIS 80

4.2.1 Hadoop and MapReduce 81

4.2.2 Mapbox API 83

4.3 Twitter API 84

4.4 The Architecture of the Implementation 86

4.4.1 Data Collection 86

4.4.2 Data Processing 88

4.4.3 Data Visualization 90

4.5 Case Study 92

4.5.1 Filters in Data Collection 92

4.5.1.1 Collected Tweets for the Study Area 93

4.5.2 Querying Tweets Using Hadoop 94

4.6 Data Visualization 96

4.7 Conclusions 100

Chapter 5 105

5 Conclusions and Future Work 105

5.1 Internet Computing 106

5.2 Cloud Computing 107

5.3 Research Limitations 108

5.4 Research Contribution 108

5.5 Future Work and Recommendation 109

Curriculum Vitae

LIST OF TABLES

Table2.1: Results of Implementation 42

Table 3.1: Summary of Cloud Computing Services 60

Table 3.2: Summary of cloud deployment models 63

LIST OF FIGURES

Figure1.1: Internet Computing Implementations Workflow 7

Figure1.2: Cloud Computing Workflow 8

Figure2.1: WPS Process (OGC 2008).....17

Figure 2.2: 52°North Architecture (http://www.52north.org).....26

Figure 2.3: ZOO Components (http://www.zoo-project.org) 28

Figure 2.4: Deegree Architecture (http://www.deegree.org/) 31

Figure 2.5: PyWPS Architecture (http://pywps.wald.intevation.org) 33

Figure 2.6: Crop Process Results 39

Figure2.7: Landsat Images for the Study Area 40

Figure 3.1. Evolution of Computing Platforms (Hand book of cloud, 2010) 50

Figure3.2 Cloud Computing Architecture 58

Figure 3.3: Public Cloud Model.....61

Figure 3.4: Private Cloud Model.....61

Figure 3.5: Hybrid Cloud Model.....62

Figure 4.1: Overview of the General Architecture of the Implementation 86

Figure 4.2: Architecture of the Data Collection Component 87

Figure 4.4: Hadoop Ecosystem with Hive and Hbase.....90

Figure 4.5: Data Visualization Component Architecture.....91

Figure 4.6: Time of Query Execution on the Whole Dataset Using Hive on top of Hbase 95

Figure 4.7: SQL Query in Hive for Retrieving Words Ending in ING.....96

Figure 4.8: Overview of Displayed Tweets on Top of Base Map in Vancouver Area 97

Figure 4.9: Possible Hotspots of Grouse Mountain 98

Figure 4.10: Possible hotspot of Grouse Mountain, Afternoon and Evening Tweets zoomed in 98

Figure 4.11: Morning, Afternoon, Evening, Night Tweets shown in Different Colors ...99

Figure 4.12: Tweets Along Grouse Grind Trail 100

LIST OF SYMBOLS AND ABBREVIATIONS

AJAX - Asynchronous JavaScript and XML

API - Application Programming Interface

AR - Augmented Reality

ASP - Active Server Pages

BAO - Business Analyst Online

CSS - Cascading Style Sheet

CSV- Comma-Separated Values

EDI - Electronic Data Interchange

ESRI - Environmental Systems Research Institute

GDAL - Geospatial Data Abstraction Library

GeoTIFF – Geospatial Tagged Image File Format

GIFF - Graphics Interchange Format

GIS - Geospatial Information System

GML - Geography Markup Language

GUI - Graphical User Interface

HDF - Hierarchical Data Format

HDFS - Hadoop Distributed File System

HTML - Hyper Text Markup Language

IAAS - Infrastructure As A Service

IBM - International Business Machines

IDG - International Data Group

IT - Information Technology

JAI - Java Advanced Imaging

JPEG - Joint Photographic Experts Group

JTS - Java Topology Suite

KML - Keyhole Markup Language

KVP - Key Value Pair

OGC - Open Geospatial Consortium

OGR - OpenGIS Simple Features Reference

PAAS - Platform As A Service

PC - Personal Computer

PHP - Hypertext Preprocessor

PNG - Portable Network Graphics

RDBMS - Relational Data Base Management System

REST - Representational State Transfer

SDI - Spatial Data Infrastructure

SAAS - Software As A Service

SDK - Software Development Kit

SLA - Service Layer Agreements

SOA - Service Oriented Architecture

SOAP - Simple Object Access Protocol

SQL - Structured Querying Language

TED - Technology Entertainment Design

TIFF - Tagged Image File Format

VM - Virtual Machine

WCS - Web Coverage Service

WFC - Web Feature Service

WKT - Well-Known Text

WMS - Web Mapping Service

WPS - Web Processing Service

WSDL - Web Service Definition Language

XHTML - eXtensible Hyper Text Markup Language

XML - eXtensible Markup Language

Chapter 1

1 Introduction

In this thesis, cloud computing platform and Internet computing platform have been studied to test and evaluate their capability in storing and processing big spatial data and to show the evolution of computing from Internet to cloud in terms of spatial data processing. In order to study Internet computing, Open Geospatial Consortium's (OGC) Web Processing Service (WPS) implementations have been studied. To evaluate capabilities of WPS in handling raster data (satellite imagery) as an example of big spatial data, a raster calculation algorithm has been implemented and the results have been analyzed.

To study cloud computing, a scalable platform has been developed. The platform collects, processes and maps activities of people using geolocated data collected from Twitter as example of big spatial data. The goal of this cloud computing platform is to map activities of people in various locations and to add extra information about a specific place. This helps users of the platform perceive a better understanding of that place by having the knowledge on how people are connected to that location.

As the research had been aimed at storing and processing big spatial data, the main feature of Internet computing and cloud computing platforms that needed to be tackled was scalability of them. Investigating OGC's Web Processing Service (WPS) and Web Mapping Service (WMS) implementations, it was found that OGC's standards have not addressed the issues in collecting and processing large data volumes, they can't deal with unstructured data and are not scalable when it comes to rapid growth in data size.

In evaluating cloud computing platform, cloud based technologies, models and services have been studied to determine the capability of a cloud based platform to store and process big spatial data. Furthermore, a review on evolution of cloud computing has been done, the models, frameworks and architecture of cloud computing have been explored and studied in respect with GIS. In order to test cloud computing's capability in storing and processing big spatial data, a cloud based structure has been proposed and a platform has been developed to store, process and visualize geotagged Tweets. The proposed structure has been implemented for a well-known trail, Grouse Grind, located in a park in North Vancouver, British Columbia. Throughout the study of cloud computing platform, it was found that the platform can easily be scaled to handle the flow of big spatial data (Tweets) and is able to manage unstructured data types. In the end of the research, a comparison has been made on advantages and disadvantages of cloud computing platform and Internet computing platform in regards with storing and processing of big spatial data.

1.1 Research Background

There has been a substantial amount of research conducted in developing and implementing platforms capable of handling geospatial data using OGC's standards that has mainly be focused on the role of WMS in data visualization and analysis in geospatial applications (Hwang and Luetkemeyer 2010) and the role of WPS in 3D geographical data analysis (Lanig et al., 2012). Researchers have also developed distributed applications for geospatial analysis using BPEL and WSDL as Internet

computing implementations (Meng et al., 2010). However the deficiency in OGC's implementations in handling big data (satellite images, Tweets) becomes evident when dealing with a large volume of unstructured data (e.g. Tweets) which not only needs distributed computing frameworks to increase processing and storing power but also scalable computing frameworks capable of automatically scaling up the processing power and storing power when necessary and also databases with the ability of storing and processing unstructured data.

Cloud computing solved many problems regarding data processing with the rise of big data, and as a result, there has been a wide range of research work which included the design of cloud computing architectures(Rimal et al., 2010), (Liu et al., 2011), (IBM, Breiter, 2010), (Christopher S. Yoo, 2011), the deployment of applications of cloud computing (Yang and Wu, 2010) and also big data processing in cloud computing (Ji et al., 2012), (Nepal and Pendy, 2013). Zhu (2010) provides an overview of the cloud computing ecosystem. Most of the researchers have been trying to define and use cloud computing and cloud based platforms for various applications such as natural disaster management (Habiba and Akhter, 2013). However, there has not been much research carried out in investigating cloud computing for processing of geospatial data, and the few examples in this area include efforts in implementing a cloud computing based GIS (Yiqin et al., 2011), brief overview of GIS applications in cloud computing (Aysan et al., 2012) and examples of geospatial data services in cloud computing (Wu et al., 2011).

With the growing popularity of Twitter and with the flow of geotagged tweets, many researchers started to use geotagged Tweets for spatial analysis, examples of which include community building (Shoko Wakamiya et al., 2011), behavior pattern

recognition (Fujisaka et al., 2012) and mining mobility patterns (Gabrielli et al., 2012). However, there has been a few research attempts carried out in processing geotagged Tweets using cloud enabled technologies and implementations. Moreover, few applications have used cloud computing in their analysis, some examples include urban planning applications (Piacentini, 2013) and sentiment analysis (Sundar 2012).

The first efforts of this research focused on investigating image calculation capabilities of OGC standards implementations of Web Processing Service (WPS) as Internet computing platform for processing and mapping geospatial data over Internet. Once it was tested and concluded that WPS implementations are not suitable for processing large data volumes, to show the evolution of computing in processing large volume of spatial data, cloud computing was studied to be able to find more suitable platforms to work large data volumes. In cloud computing platform, an architecture has been designed and implemented to work with large data volumes to fill the gap of Internet computing in dealing with large data volumes.

1.2 Research Challenges

In the process of conducting this research, a number of challenges have been dealt with and can be summarized as one of the following:

- **Lack of standards**: As opposed to Internet computing platform and OGC's WPS and WMS, which are well-established and published standards on collecting and sharing geospatial data, the standards for cloud computing are just not available yet. The lack of standards for cloud computing has its own set of problems when

it comes to designing and developing a cloud computing platform. Currently there is a very wide range of opinions, regarding the diversity of methods for implementing cloud computing platforms, the diversity of cloud models that can be used for the same application, the diversity of cloud services available for each platform and many other issues related to cloud computing such as scalability, privacy and virtualization.

- **Clarification of Terminology**: The next challenge in this study was the existence of a large number of definitions, often vague and sometimes contradictory, on cloud computing. The reason for this lack of consensus lies in the fact that every researcher as well as vendor of cloud technology, has tried to define cloud computing from the perspective of a particular application, research project or business interest. As a result, this has generated too many vague statements and definitions on cloud computing.
- **Handling large volume of geospatial data:** The third challenge in this thesis was to collect, process and visualize large volumes of geospatial data using the Internet and Cloud computing implementations. For Internet computing implementations, Landsat data was used and for cloud computing implementation geotagged Tweets were used. In both cases and due to lack of hardware in processing power, input data was down sized. Landsat scenes were reduces in size and geotagged Tweets were reduced in number.

1.3 Research Objectives

The main research objective of this study is to review, evaluate and test two computing implementations, cloud computing and Internet computing, in order to find a scalable platform, capable of storing and processing and visualizing large volume of geospatial data and to portray the evolution of the computing architecture from Internet computing to cloud computing.

To achieve this goal, the following objectives have been defined:

- To review and study OGC's WPS implementations as standard examples of spatial Internet computing platforms
- To study WPS implementations for raster processing of large volumes of Landsat satellite images.
- To evaluate WPS implementations for their scalability in processing these images.
- To review cloud computing, its models, services, architectures and related technologies.
- To study cloud computing platform for storing and processing and visualizing geotagged tweets.
- To evaluate both Internet computing and cloud computing implementations for their ability in handling large volumes of geospatial data.
- To show the evolution of computing among the two implementations in terms of scalability, storing, processing and visualizing large volumes of geospatial data.

1.4 Methodology

The methodology consists of two approaches. In the first approach, Internet computing has been investigated to see if WPS implementations are suitable for handling large data volumes and then in the second approach cloud computing was evaluated for the same question. To accomplish the objectives of the thesis, a brief description of the research methodology is provided in this section in the form of a flowchart.

Internet computing platform:

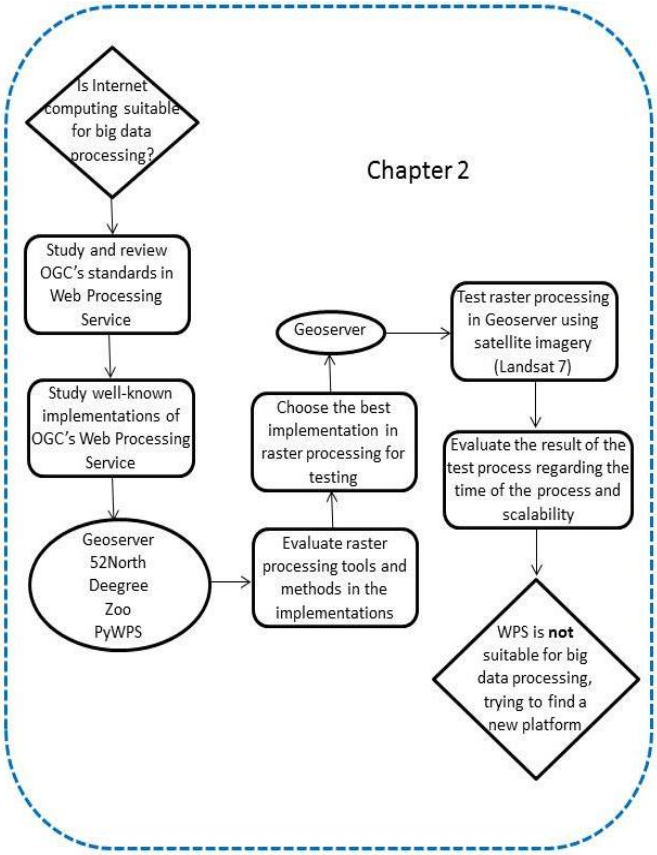

Figure1.1: Internet Computing Implementations Workflow

Cloud computing:

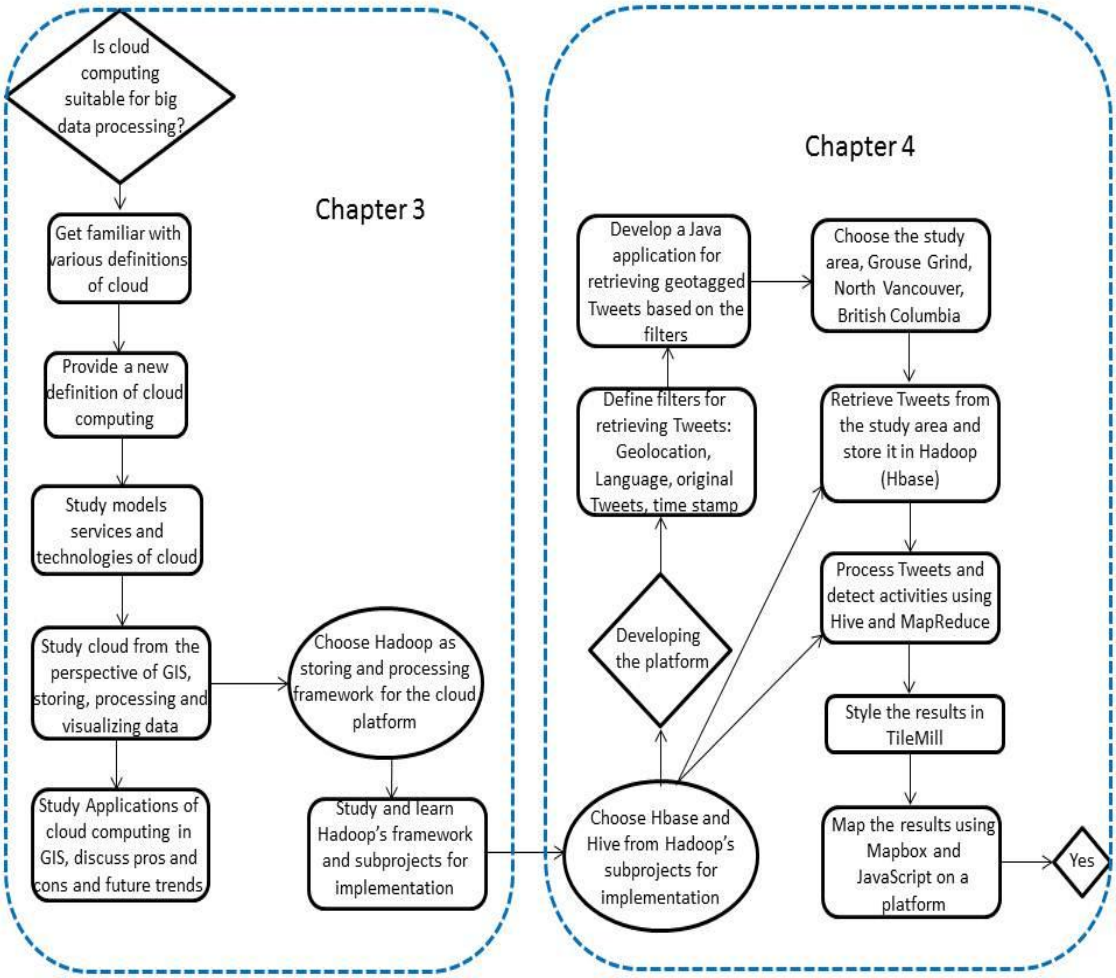

Figure1.2: Cloud Computing Workflow

1.5 Overview of the Chapters

The remainder of this thesis is organized as follows.

Chapter 2 gives an overview of the OGC's Web Processing Service including the possible processes in WPS, the supported data types in WPS, the architecture of WPS and its advantages and disadvantages in respect with handling large volume of imagery

data. Furthermore Geoserver, 52North, Deegree, Zoo and PyWPS have been studied for their capability in raster calculation, implementation issues and considerations of WPS has been elaborated and in the end the most efficient WPS implementation being Geoserver in raster calculation has been introduced. The content of this chapter was presented during the joint Canadian Institute of Geomatics Annual Conference and the 2013 International Conference on Earth Observation for Global Changes (EOGC'2013), 5-7 June 2013, Toronto, Ontario, Canada.

Chapter 3 presents the evolution of cloud computing from distributed computing and grid computing. Furthermore in this chapter, cloud computing is described from a GIS perspective, and a summary of the main impacts of cloud computing on storing, processing and visualizing geospatial data is provided using a number of applications as examples of successful cloud based applications. Hadoop and MapReduce projects are explained in more detail since Hadoop has been chosen for implementing the cloud platform of this study. Sections 3 and 4 from this chapter have been presented in the Geomatics Atlantic conference in Saint John New Brunswick in September 2013. This Chapter will be submitted to the Geography Compass Journal as a peer review paper.

Chapter 4 describes the designed and implemented cloud platform for collecting, processing and visualizing geotagged tweets. A Java based application has been developed to collect geotagged tweets, The Hadoop project has been used along with Hbase and Hive as sub projects of Hadoop in order to facilitate a near-real time processing of geotagged tweets sent about their visit to the Grouse Mountain Park in Vancouver British Columbia. The JavaScript and Mapbox library were then used for

generating a dynamic base map of the location of the visitors' activities in the park. This chapter will be submitted to Transactions in GIS (peer-reviewed journal)

Finally, Chapter 5 concludes and summarizes the contribution of this study and provides recommendations for future research.

Reference

Aysan, I., Yigit, H. and Yilmaz, G. (2011), 'GIS applications in cloud computing platform and recent advances', In 2011 5th International Conference on Recent Advances in Space Technologies (RAST), 2011, pp. 193–196.

Breiter, G., (2010), 'Cloud Computing Architecture and Strategy', IBM Corporation, available at http://www.minet.uni-jena.de/dbis/lehre/ss2010/saas/material/ibm-breiter.pdf

Fujisaka, T., Lee, R. and Sumiya, K. (2010), 'Monitoring geo-social activities through micro-blogging sites', Proceedings of the 15th international conference on Database systems for advanced applications, April 01-04, 2010, Tsukuba, Japan, pp: 13-18

Gabrielli, L., Rinzivillo, S., Ronzano, F. and Villatoro, D. (2013). 'From Tweets to Semantic Trajectories: Mining Anomalous Urban Mobility Patterns', In J. Nin & D. Villatoro (eds.),CitiSens(pp. 26-35), : Springer. ISBN: 978-3-319-04177-3

Habiba, M. and Akhter, S. (2013), 'A Cloud Based Natural Disaster Management System' Grid and Pervasive Computing Volume 7861, 2013, pp 152-161

Hwang, A. and Luetkemeyer, K., (2010), 'Using Web Map Service Data for Visualization and Analysis in Geospatial Applications', MathWorks News&Notes at http://www.mathworks.com/tagteam/65098_91843v00_NN10_FA_WebMap.pdf

Ji, C., Yu, L., Wenming, Q., Awada, U. and Li, K., (2012), 'Big Data Processing in Cloud Computing Environments', 12 International Symposium on Pervasive Systems, Algorithms and Networks, December 2012

Lanig, S. and Zipf, A., (2010), 'Proposal for a Web Processing Services (WPS) Application Profile for 3D Processing Analysis' 2nd International Conference on Advanced Geographic Information Systems, Applications, and Services (GEOProcessing 2010), St. Maarten, Netherlands Antilles, 10-15 February 2010, pp. 117-122

Liu, F., Tong, J., Mao, J., Bohn B., Messina V., Badger M. and Leaf, D., (2011). 'NIST Cloud Computing Reference Architecture', The National Institute of Standards

and Technology (NIST) Publications, NIST SP - 500-292, 8 September 2011, pp.35

Meng, X., Xie, Y. and Bian, F., (2010), 'Distributed geospatial analysis through web processing service: a case study of earthquake disaster assessment', Journal of Software, 5(6), pp. 671-679.

Pandey, S. and Nepal, S. 'Editorial: Cloud Computing and Scientific Applications – Big Data Analysis in the Cloud', The Computer Journal, Oxford Press (in progress)

Piacentini, A. (2013), 'Could Twitter help urban planners improve transport networks?', Urban Sensing project, 25 March 2013, available at http://urban-sensing.eu/?p=628

Rimal, PB., Jukan, A., Katsaros, D. and Goeleven, Y., (2011) 'Architectural Requirements for Cloud Computing Systems: An Enterprise Cloud Approach', Springer Science Business Media B.V, Journal of Grid Computing (2011), Volume 9, pp.3–26

Sundar, J. (2012), 'A Real Time Sentiment Analysis Application using Hadoop and HBase in the Cloud', At http://blogs.wandisco.com/2012/06/30/a-real-time-sentiment-analysis-application-using-hadoop-and-hbase-in-the-cloud/

Wakamiya, S., Lee, R. and Sumiya, K. (2012), 'Looking into Socio-cognitive Relations between Urban Areas based on Crowd Movements Monitoring with Twitter', DBSJ Journal, Vol. 11, No. 2, pp. 19-24, October 2012

Wu, B., Wu, X., and Huang, J. (2010), 'Geospatial data services within Cloud computing environment', International Conference on Audio Language and Image Processing (ICALIP), 2010, Nov. 2010, pp 1577 - 1584

Yang, J. and Wu, S. (2010), 'Studies on Application of Cloud Computing Techniques in GIS', Second IITA International Conference on Geoscience and Remote Sensing , 978-1-4244-8515-4/10/ 2010 IEEE, pp.492-495

Yiqin, L., Kanghua, Y. and Yuan, L. (2011) , 'An Implementation of Embedded Geographic Information System Based on Cloud Computing', Third Pacific-Asia Conference on Circuits, Communications and System (PACCS).

Yoo, C.S., (2011), ‘Cloud Computing: Architectural and Policy Implications’, Faculty Scholarship, Paper 358. At: http://scholarship.law.upenn.edu/faculty_scholarship/358

Zhu, J. (2010), ‘Cloud Computing Technologies and Applications’, Hand book of cloud computing, chapter 2, pp21, 45, ISBN: 978-1-4419-6523-3

Chapter 2

2 A Comparative Study of Web Processing Service Implementations for Raster Calculation

Abstract

The reliability of Geographic Information Systems (GIS) and the systematic use of the Open Geospatial Consortium Web Services (OWS) have led to a variety of technologies and methods to store and process geospatial data over the Internet. There are a number of Web Processing Service (WPS) implementations for geospatial data processing: each one having a set of features, specifications and flexibilities focused on a specific aspect of WPS, such as image processing in Geoserver or Python based processing in PyWPS In this Chapter the well-known WPS implementations, ZOO, 52° North, Geoserver, Deegree and PyWPS will be compared in terms of their capabilities in handling raster calculations. The comparison will include the input and output data format capabilities of these implementations, their out-of-the-box analysis capabilities, their flexibility and power to add new raster analysis and calculations, their compatibility with existing GIS platforms and their main services. At the end of this chapter the most optimized WPS implementation for raster processing will be selected.

2.1 Introduction

Standardization of geospatial data and metadata has become crucial in the context of development of GIS due to OGG's specific directives and policies regarding geospatial data use and sharing. As a result, numerous applications are available today to store and access geospatial data over the Internet using Web Map Services (WMS), Web Feature Services (WFS) and Web Coverage Services (WCS). Many open source or proprietary GIS solutions currently support standards to access or modify geospatial data (Bocher 2009), but only a few are available for processing such data through Web Processing Service (WPS).

The OGC WPS specification provides the service interface definitions to specify a wide range of processing tasks as geospatial web services in order to distribute common GIS functionalities over the Internet (Friis-Christensen et al., 2009; Li et al., 2010). The main difference between desktop functionalities and WPS services is that the latter can be accessed remotely and assembled in varied web integration scenarios (Brunner et al., 2009; Lowe et al., 2009). Some examples of the functionalities include image classification, image processing and mosaicking.

The OGC WPS also provides access to calculations or models that operate on spatially-referenced data, which can be available locally, or delivered across a network using download services such as WFS (Web Feature Services), WCS (Web Coverage Services) and SOS (Sensor Observation Services). While most OGC specifications and standards are devoted to geospatial data models and access, the OGC WPS specification is focused on processing heterogeneous geospatial data. The typical steps consist of the

identification of spatially-referenced data required, execution of the process, and the management of the output process by client applications.

This Chapter describes a comparative study of five of OGC's WPS implementations. The first section of the paper explains the concept of WPS as well as the ways in which WPS works and operates, the advantages and disadvantages of WPS and some key applications for WPS. In the next section, Geoserver, 52° North, ZOO, Deegree and PyWPS are compared based on their raster calculation features and capabilities. Finally, the Chapter presents the practical results obtained from these different implementations and the recommendations for future research work.

2.2 Definition of WPS

WPS is one of the most recent interoperability standards published by Open Geospatial Consortium (OGC, 2012). It was first proposed under version 0.4 in 2005 (OGC 2005), and some improvements were added in version 1.0.0, which was released in 2007 (OGC 2007a, b).

WPS is designed to standardize the way that GIS algorithms are made available through the Internet. It specifies a means for a client to request the execution of a spatial calculation from a service. It intends to automate geoprocessing by employing geospatial semantics in a Service Oriented Architecture (SOA). WPS supports simultaneous processes via the HTTP GET and POST method, as well as the Simple Object Access Protocol (SOAP) and Web Services Description Language (WSDL). As a result, WPS

offers a simple web-based method of finding, accessing, and using all kinds of calculations and models (Figure 2.1).

WPS also defines a standardized interface that facilitates the publishing of geospatial processes, and the discovery of and binding to those processes by clients. Processes can include any algorithm, calculation or model that operates on spatially referenced data. Publishing means making available machine-readable binding information as well as human-readable metadata that allows service discovery and use. The WPS interface standardizes the way processes and their inputs/outputs are described, how a client can request the execution of a process, and how the output from a process is handled. WPS uses standard HTTP and XML (eXtensible Markup Language) as a mechanism for describing processes and the data to be exchanged.

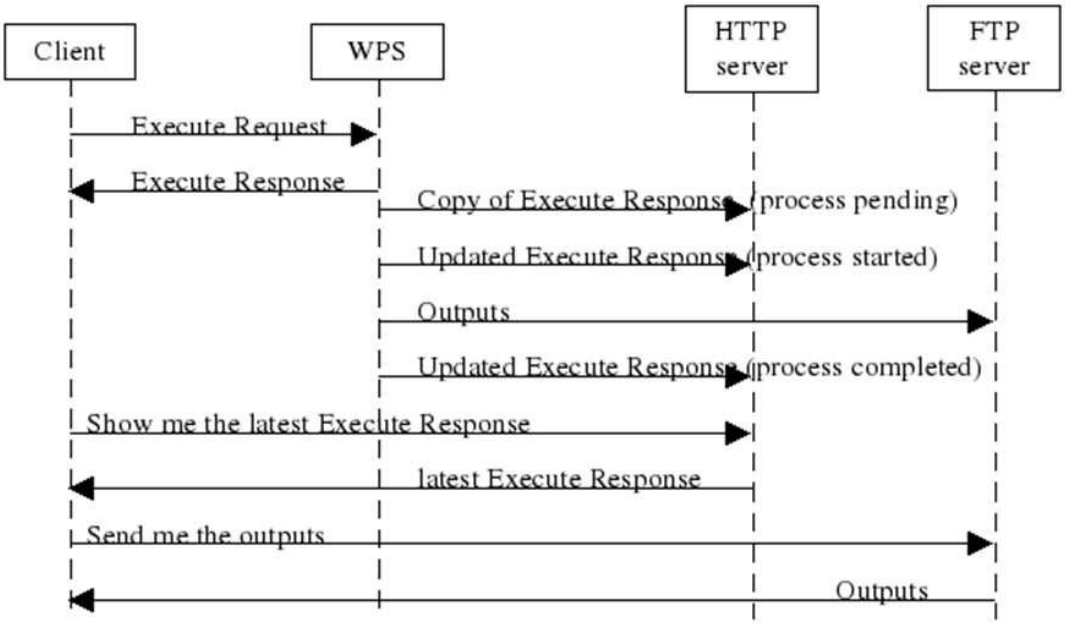

Figure2.1: WPS Process (OGC 2008)

2.2.1 Processes of WPS

It is important to point out that WPS does not specify the type of processes that could be implemented as a web service. Instead, it specifies a generic mechanism that can be used to describe and web-enable any sort of geospatial process. Therefore, a WPS can be configured to offer any sort of GIS functionality to clients across a network, including access to pre-programmed calculations and/or computation models that operate on spatially referenced data. The calculations can be extremely simple or highly complex, with any number of data inputs and outputs. A WPS may offer calculations as simple as subtracting one set of spatially referenced numbers from another (e.g., determining the difference in influenza cases between two different seasons), or as complicated as a global climate change model.

2.2.2 WPS Operations

There are three mandatory requests that can be submitted to a WPS server: GetCapabilities, DescribeProcess and Execute. First, GetCapabilities provides a capability document containing important metadata information about the WPS Server instance and a list of the services available on the server-side presented as a unique identifier, a title and a short description. Second, Describe Process provides more detailed information on one or more services, containing the necessary input data, the targeted output data format, as well as a title and a short abstract for each of these entries. Once all the necessary parameters are gathered from the DescribeProcess request, the processing task can be submitted to the server using the Execute request.

The latter can answer directly to the client by returning the created output or storing it as a Web accessible resource (OGC 05-007r4, 2005).

Each implementation of WPS defines the processes that it supports, as well as their associated inputs and outputs. WPS can be thought of as an abstract model of a web service, for which profiles need to be developed to support use, and standardized to support interoperability. As with the other OGC specifications GML and CAT, it is the development, publication, and adoption of profiles which define the specific uses of this specification.

The WPS discovery and binding mechanisms follow the OGC model set by WMS and WFS, in that WPS defines a GetCapabilities operation, and requests are based on HTTP Get and Post (OGC 07-063r1). WPS does more than just describe the service interface, in that it specifies a request/response interface that defines how to:

- encode requests for process execution,
- encode responses from process execution,
- embed data and metadata in process execution inputs/outputs,
- reference web-accessible data inputs/outputs,
- support long-running processes,
- return process status information,
- return processing errors,
- request storage of process outputs,
- create a service chain.

2.2.3 Data Types in WPS

WPS is targeted at processes that involve geospatial data (vector and raster), but can also be applied to non-spatial processes as well. The data required by the WPS can be delivered across a network, or available at the server. WPS defines three types of data as one of the following:

- Complex Data: includes imagery, XML, CSV, and custom or proprietary data structures.
- Literal Data: includes single numerical values or text strings.
- Bounding Box Data: includes geographic coordinates for a rectangular area.

WPS does not define or restrict the type of data required or output by a process. Instead, it identifies a generic mechanism to describe the data inputs required and produced. Thus data can include image data formats such as GeoTIFF, or data exchange standards such as GML. Data inputs can also be legitimate calls to OGC web services. For example, a data input for an intersection operation could be a polygon delivered in response to a WFS request, in which case the WPS data input would be the WFS query string.

2.2.4 Advantages of WPS

The main advantage of WPS is interoperability of network-enabled data processing. It allows organizations to deliver calculations to users independent of the underlying

software. This independence helps to ensure the longevity of code. Further benefits include:

- Support of multiple web service approaches: it defines equivalent KVP Get, XML Post, and SOAP interfaces, allowing the user to choose the most appropriate interface.
- Power of distributed computing: WPS is designed to enable distributed processing of geospatial data located anywhere on the Internet.
- Fast, reliable access to "near real-time" calculations: because WPS makes calculations available as web services, the most up-to-date data can be accessed directly from the source organization responsible for its maintenance.
- Reusability: exposing processes through WPS allows organizations to reuse the same process (and code) in multiple applications.
- Flexibility: exposing processes through WPS allows organizations to change their underlying software without impacting users and client applications.
- Security: access to WPS processes can be restricted using a variety of standard web service security mechanisms such as firewalls and HTTPS.

WPS holds great promise for using computational tools without traditional concerns such as distributing bug fixes. However, some geoprocessing operations can be completed more effectively locally (i.e. on a user’s desktop PC) than remotely (i.e. on a central server), because of the time constraint to upload input data or transfer it from another server and subsequently download resulting outputs (Michaelis and Ames 2008).

Computational complexity plays a large part; if the process takes several hours to complete even on a small dataset, it can be better to process the data remotely. Some of this time delay may be offset by allowing a request to a WPS server to specify an alternate data source rather than a direct upload, (e.g., a Web Feature Service GetFeature request). In this case, the WPS would download the required data from another server rather than receiving it from the WPS client. Such a process could become common in service-oriented architectures.

Remote processing is also appropriate for deployment of new algorithms and code that is under active development. The traditional view of remote processing requires that input data is uploaded, as in Remote Procedure Calls where input data and parameters are sent together with a function call. However, input data could also be stored on the server, requiring the client only to specify the particular input which is desired. This creates the opportunity for a WPS server to also serve raw or pre-processed data. This approach could be particularly useful for processes requiring real-time data such as weather station observations or live traffic observations. WPS services could be provided by the same entity that collects the data, allowing the processes to have access to the latest available data at all times. The motivations for using a remote processing server are many, but ultimately the decision must lie with the user whether remote processing is appropriate for the task (Michaelis and Ames 2008).

Considering this point, WPS allows several different approaches for executing a process:

- Returning raw outputs: The simplest approach is only applicable when the WPS produces only one output. In this case, the output can be returned directly to the

users in its raw form. For example, a request to buffer a feature could return an image of the buffered feature encoded in a PNG file.

- Returning outputs embedded in XML: One response to an Execute request is an XML document that includes metadata about the request, as well as the outputs from the process encoded and wrapped in the XML response. This form of response is recommended when the output size is less than a few megabytes in size, and the user requires metadata found in the wrapper.
- Storing outputs: A WPS may allow a user to request storage of the outputs. In this case, the XML document returned to the client will again contain metadata, but instead of the outputs themselves, it will contain references to web-accessible locations from which the outputs can be downloaded.
- Long-running processes: Finally, if an Execute request triggers a long-running process, the WPS will return a response containing references to the outputs as indicated above. Also included will be a reference to a location where the Execute response document is stored. The WPS will periodically update the status indicator found in this document until processing is complete.
- Providing access to data produced by a WPS: The outputs from a WPS are available to the client that initiated the Execute operation. The specification does not address the archival, cataloguing, discovery, or retrieval of WPS outputs, so that other clients can access them through Email, Client/Server index, or a registry.

2.3 WPS Implementations

In this section, the five WPS implementations are described, based on their raster analysis capabilities as an example of geoprocessing.

2.3.1 52°North Project

The 52°North (52 North, 2012) Web Processing Service enables the deployment of geo-processes on the web in a standardized way. It features a pluggable architecture for processes and data encodings. 52°North's focus is on the creation of an extensible framework to provide algorithms for generalization on the web (52North Geoprocessing, 2012).

2.3.1.1 Features of 52°North

The general Features of this implementation include following:

- A full Java-based Open Source implementation.
- Support to all features and operations of WPS specification
- A pluggable framework for algorithms and a XML data handling and processing framework.
- Build-up robust libraries (e.g. JTS, GeoTools, XMLBeans, servlet API, derby)

The 52°North WPS also supports a wide range of standards and data types, supporting SOAP, WSDL, HTTP-GET and HTTP-POST among the standards and for data types it supports raw data types, GeoTiff, ArcGrid, GML, Shapefiles, KML and WKT.

One of the main advantages of 52°North is the range of extensions it supports, making possible a wide range of processing to be performed. The extensions include WPS4R - R Backend,GRASS out of the box extension, 220+ SEXTANTE Processes and ArcGIS Server Connector.

Having these libraries and processes as backend, 52°North has become one of the most used WPS implementations as it supports all kinds of process. 52°North is currently seeking to add Udig, Jump and open layers as its clients as well to further enhance its performance.

2.3.1.2 52°North Architecture

The 52°North WPS supports a set of input formats, processes and output formats as discussed above. The architecture is held pluggable to enable the extension of already supported formats and processes. In detail, the parsers (green arrows) transform external data formats (such as GML, SHP, etc.) into internal data formats (Figure 2.2).

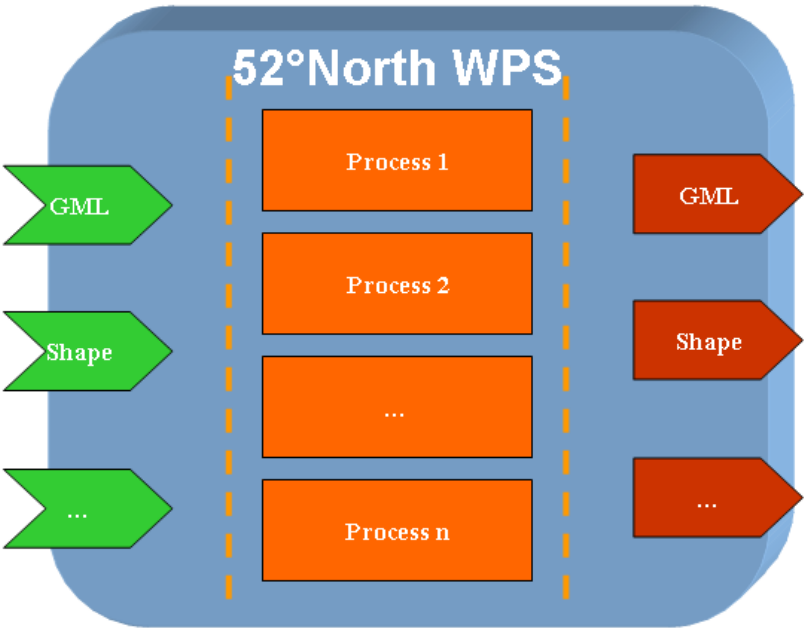

Figure 2.2: 52°North Architecture (http://www.52north.org)

The output of the process can be stored and accessed using various approaches such as one of the following:

- All Results can be stored as simple web accessible resource with an URL
- Raster/Vector results can be stored directly as WMS layer
- Vector results can be stored directly as WFS layer
- Raster results can be stored directly as WCS layer

This variety in dealing with output results has made 52°North more flexible to use as the output can be accessed in different forms and as a result the output can be displayed in more applications based on the need of users (52North Geoprocessing, 2012).

2.3.1.3 Raster Calculations in 52°North

As 52°North can connect to multiple libraries and can be paired with different extensions, it has a wide range of support to raster formats and raster analysis algorithms, for instance, GRASS backend supports Tiff, GeoTiff, JPEG, PNG and GIF as input and output in raster calculations, WPS4R backend supports GRASS formats as well as DBF, NetCDF and application image formats. Adding ArcGIS to 52°North and hundreds of processes and algorithms and the data types being supported by ArcGIS makes 52°North the choice of many developers and researchers for geoprocessing over Internet (52North Geoprocessing, 2012).

It should be noted that connecting to extensions as backend and running the algorithms over a network or Internet might be a time consuming process depending on the network/Internet speed, the server's work load and the algorithm itself, as a result, in-built and out of the box calculations seem to be more suitable for real-time processing. 52°North currently supports 20 out of the box vector processing and 6 raster processing, but using the discussed backends and 52°North's WPS SDK, custom processes can be developed and incorporated in it.

2.3.2 ZOO Project

ZOO-Project, is a new open source implementation of the OGC Web Processing Service (WPS), released under the term of the MIT/X-11 license. This license means that anyone can use, copy, modify, distribute and publish. ZOO is based on a server-side C language Kernel (ZOO Kernel), ZOO-Project proposes a new approach to develop,

handle and chain standardized GIS-based Web services. Therefore, it is made of three parts: ZOO Kernel, ZOO Services, ZOO API (Figure2.3).

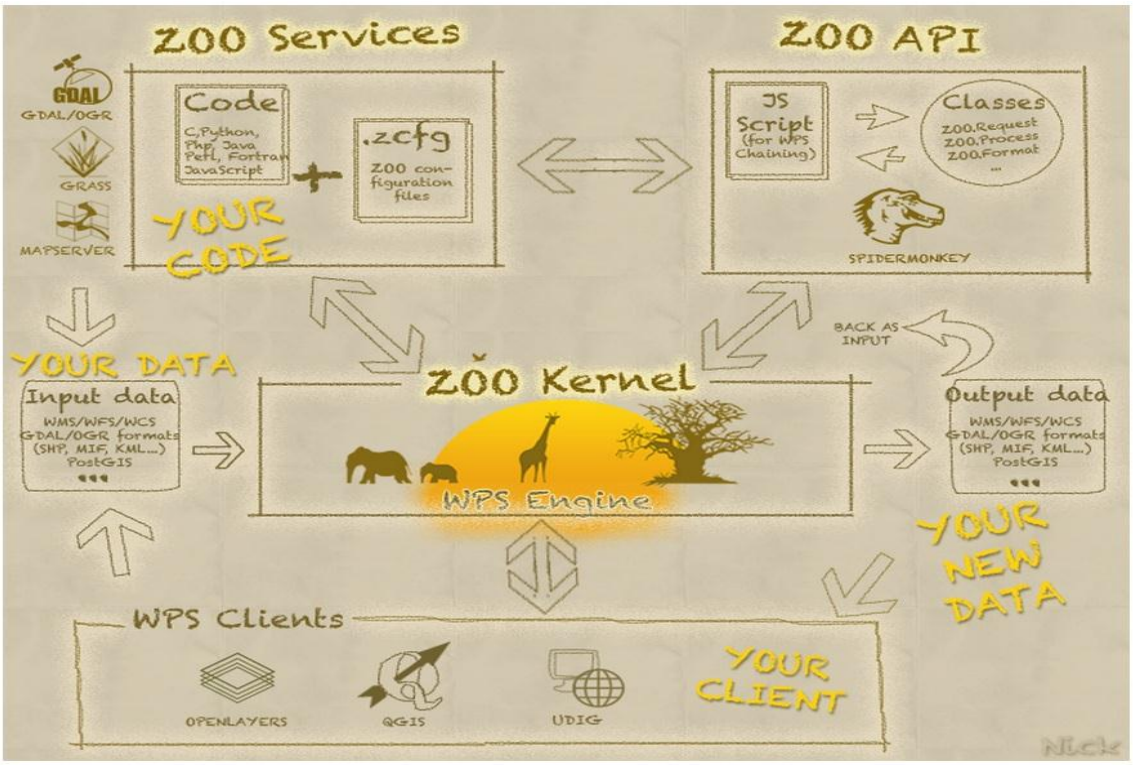

Figure 2.3: ZOO Components (http://www.zoo-project.org)

The ZOO-Project offers an alternative WPS implementation that supports multiple programming languages namely Java, JavaScript, Python and PHP and simplifies the development of new services as independent modules. Nevertheless, ZOO Kernel still lacks the WSDL and SOAP support which is required for WPS Servers. The ZOO-API provides to services developers a way to build complex services using already existing ones by chaining them and adding logic in the chain.

2.3.2.1 Raster Calculations in ZOO

ZOO only offers 23 separate functioning processes after installation with just five Raster calculations out of the box, with the raster calculations including the creation of sample rasters, converting rasters with different formats and three other basic raster operations. In contrast, ZOO-Kernel is able to load dynamic libraries: one of the libraries which is a good source is GDAL/OGR that can be loaded with ZOO and also ZOO can be paired with GRASSGIS which provides advanced GIS processing algorithms. In fact, GRASS 7 now provides a WPS process description exporter, which returns XML documents describing the GRASS functions (Gebbert 2009).

It is important to note the advantages of out of the box processes. First, for loading libraries into implementations, custom coding is required. As a result, using out of the box processes reduces the effort of writing and modifying codes. Second, it is faster to deploy and run an algorithm using out of the box processes which helps significantly when having large volume of data and complex algorithms (ZOO project, 2012).

2.3.3 Deegree Project

Deegree is a Java Framework offering the main building blocks for Spatial Data Infrastructures (SDIs). Its entire architecture is developed using standards of the Open Geospatial Consortium (OGC). Deegree encompasses OGC Web Services as well as clients. It is Free Software protected by the GNU Lesser General Public License (GNU LGPL) which means that the framework is free of charge and open source and users

have the freedom to run the application, find out how it works and make changes to it as well as redistribute copies of modified versions.

The components for geospatial data management include data access, visualization, discovery and security and they also include the OGC Web Map Service (WMS) reference implementation, a fully compliant Web Feature Service (WFS) as well as packages for Catalogue Service (CSW), Web Coverage Service (WCS), and Web Map Tile Service (WMTS) and Web Processing Service (WPS).

The Deegree's Web Processing Service offers good flexibility in terms of its configuration and output formats but it tends to be more difficult to work with as the WPS server API is – unlike data oriented services like WFS and WCS or portrayal oriented services like WMS and WPS – mainly geared to software developers as it involves custom coding in Java language (Degree documentation, 2012).

2.3.3.1 Deegree Architecture

Figure 2.4 shows the overall architecture of the components involved in the Deegree WPS. The HTTP interface is realized by a servlet that has to be registered into a servlet engine like Tomcat or Jetty. The servlet chooses a handler class depending on the incoming request that delegates it to the responsible service (in this case the WPService). Depending on the requested processes the WPService decides which process is responsible for handling it.

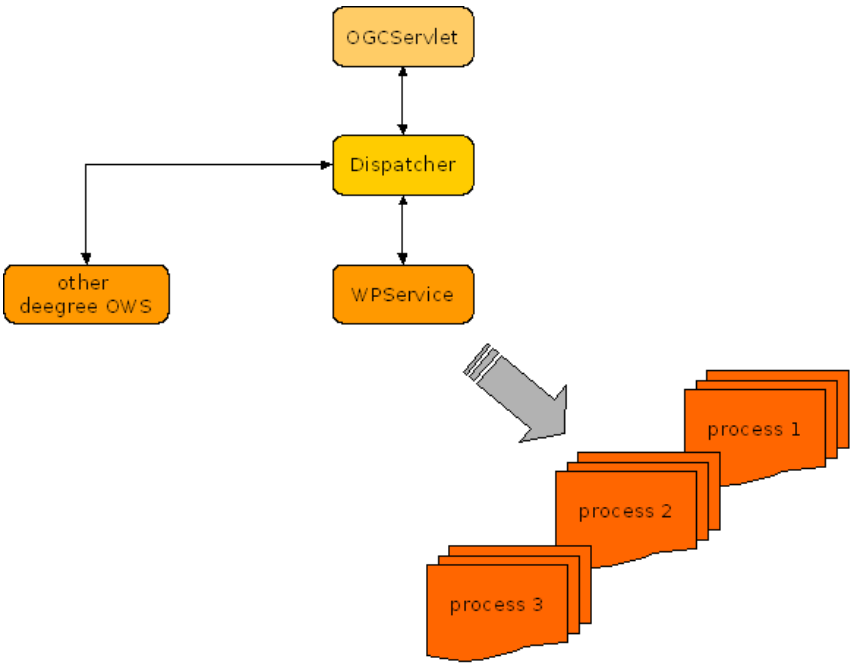

Figure 2.4: Deegree Architecture (http://www.deegree.org/)

Some important characteristics of Deegree WPS include its support of KVP, XML and SOAP requests involving all variants of input/output parameters: literal, bounding box, complex (binary and xml), streaming access for complex input/output parameters, processing of huge amounts of data with minimal memory footprints, support for storing of response documents/output parameters, input parameters given inline and by reference and asynchronous execution.

2.3.3.2 Raster Calculation in Deegree

Deegree supports more than 100 process and vast variety of vector calculations but no raster calculation ability is supported out of the box. Java language can also be used to write any type of processes as Deegree supports a good range of input/output formats.

2.3.4 PyWPS Project

Python Web Processing service (PyWPS) is an implementation of OGC Web Processing Service (OGC WPS) on the server-side. It started in 2006 and is completely written in Python language. The main advantage of PyWPS is that it has been written with native support for GRASS and as a result accessing GRASS modules via web interface is very easy (PyWPS, 2013).

2.3.4.1 PyWPS Architecture

It should be noted that PyWPS is not an analytical tool or engine. It does not perform any type of geospatial calculation. PyWPS is not a special XML parser or generator and it does not validate GMLs against given schemas and it does not build GML from Python objects. PyWPS is just a thin layer (wrapper) between the Internet and a given processing tool and as result it is not complicated. Figure 2.5 shows its architecture. As a processing tool, various popular GIS programs can be used and are supported by PyWPS such as: GRASS GIS, GDAL/OGR, Proj4 and project R (PyWPS, 2013)

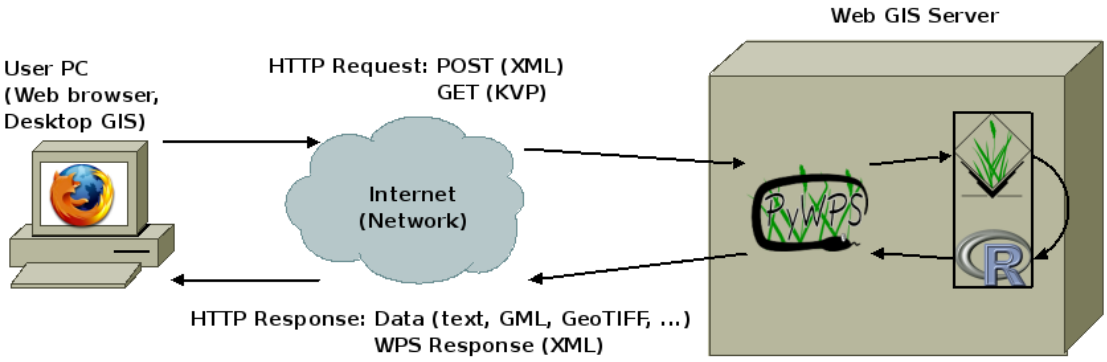

Figure 2.5: PyWPS Architecture (http://pywps.wald.intevation.org)

2.3.4.2 PyWPS Raster Calculations

There are no vector or raster processes out of the box within PyWPS, therefore users have to code custom ones. However PyWPS supports different raster formats as input as well as vector formats. There are many ways that PyWPS can improve in its future versions. The first feature is to support WSDL and SOAP, this will add more flexibility to it. Furthermore closer integration with GRASS GIS can be achieved, and better input check (xml schemas) and input custom client (OpenLayers) can be added.

2.3.5 Geoserver

Geoserver (Geoserver, 2012) is the most comprehensive implementation of WPS and also for raster calculations using WPS. The reasons are the following.

- WPS is not a part of GeoServer by default, but is available as an extension and as a service in Geoserver. As previous implementations, GeoServer is basically a Java-based software server that allows users to view and edit geospatial data.
- The main advantage of GeoServer WPS over a standalone WPS (previous examples) as stated by Geoserver foundation is its direct integration with other GeoServer services and the data catalog. This means that it is possible to create processes based on data served in GeoServer, as opposed to sending the entire data source in the request. It is also possible for the results of a process to be stored as a new layer in the GeoServer catalog. In this way, WPS acts as a full remote geospatial analysis tool, capable of reading and writing data from and to GeoServer. This feature will be discussed more on next section.

2.3.5.1 WPS Processes in Geoserver

GeoServer implements processes which can be distinguished into JTS Topology Suite processes and GeoServer-specific processes. The JTS Topology Suite is a Java library of functions for processing geometries in two dimensions. JTS conforms to the Simple Features Specification for SQL published by the Open Geospatial Consortium (OGC), similar to PostGIS. JTS includes common spatial functions such as area, buffer, intersection, and simplify.

The GeoServer WPS includes a few processes created especially for use with GeoServer. These are usually GeoServer-specific functions, such as bounds and

reprojection. They use an internal connection to the GeoServer WFS/WCS, not part of the WPS specification, for reading and writing data.

2.3.5.2 Process Chaining in Geoserver

One of the benefits of WPS is its native ability to chain processes. Much like how functions can call other functions, a WPS process can use as its input the output of another process. Many complex functions can thus be combined in to a single powerful request (Geoserver Documents, 2013)

The sequence of processes determines how the WPS request is built, by embedding the first process into the second, the second into the third, etc. A process produces some output which will become the input of the next process, resulting in a processing pipeline that can solve complex spatial analysis with a single HTTP request. The advantage of using GeoServer's layers is that data is not being shipped back and forth between processes, resulting in very good performance.

"For example, one can run the "JTS:union" process on a collection of geometries to output a single geometry that is the union of them. Processes can be chained, so one can run the "gs:Reproject" process to reproject a raster image to a different SRS, then take the output of that and run "gs:CropCoverage" to crop the raster down to a certain bounds. The result can be fed into the "gs:Import" process to save the resulting coverage as a new layer in GeoServer, for use by other clients" (http://suite.opengeo.org/opengeo-docs/processing/intro.html).

2.3.5.3 Raster Data in GeoServer

The standard GeoServer installation supports the loading and serving of Grid Coverages and Web Map Services. A coverage is a collection of spatially located features. A GridCoverage is a special case of Coverage where the features are rectangles forming a grid that fills the area of the coverage. There are many kinds of grid coverage file formats. Some of the most common are:

- **World plus image:** a normal image format like JPEG or PNG that has a side file describing where it is located as well as a PRJ side file defining the map projection just like a Shapefile. It should be noted that although the JPEG format is common due to small download size; the performance at runtime is terrible as the entire image needs be read into memory. Formats such as TIFF do not have this limitation,
- **GeoTiff:** a normal TIFF image that has geospatial information stored in the image metadata fields. This is generally a safe bet for fast performance; especially if it has been prepared with an internal overlay (which can be used when zoomed out) or internal tiling (allowing for fast pans when zoomed in).
- **JPEG2000:** the sequel to JPEG that uses wavelet compression to handle massive images. The file format also supports metadata fields that can be used to store geospatial information. There are also more formats such as ECW and MRSID that can be supported by installing the Imageio-Ext project into the JRE.
- **Web Map Service**: Another source of imagery is a Web Map Service (WMS). The Web Map Service specification is defined by the OGC. Web Map Service

Interface Standard (WMS) provides a simple HTTP interface for requesting geo-registered map images from one or more distributed geospatial databases. A WMS request defines the geographic layer(s) and area of interest to be processed. WMS is not an image file format and is able to accommodate the above mentioned image formats.

At a basic level information from a WMS can be fetched using a GetMap operation also WMS Service offers a GetCapabilities document that describes what layers are available and what other operations like GetMap are available to work on those layers. As a part of GeoServer, WMS is supported and WPS can use WMS as an input or output for a process.

2.3.5.4 Raster calculation in Geoserver

Currently there are more than 45 in-built vector process in Geoserver and more than 15 out of the box raster process. Sextante can be added to Geoserver as a backend like 52North to add more than 200 raster processes. Using Geoserver's SDK, custom processes and algorithm can also be written in Java language. Below as an example, "Crop" process on a sample raster provided by Geoserver is carried out.

In a Crop process a coverage will be cropped based on the specified cutting geometry. As can be seen in the execute document of the crop example, the coverage has coordinates ranging from the lower corner (-180, -90) to upper corner (180, 90) . therefore a polygon with 4 cordinates as its vertexes (-19 48, 20 65, 46 48, 20 30) is defined to crop the coverage. The identifier of the process is called "cropshape" and the

output format of the cropped coverage has also been defined as “TIFF”. the execution of this process will give the result as shown in Figure 2.6.

Crop example:

```
<?xml version="1.0" encoding="UTF-8"?>
<wps:Execute version="1.0.0" service="WPS" ...>
  <ows:Identifier>gs:CropCoverage</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>coverage</ows:Identifier>
      <wps:Reference mimeType="image/tiff" xlink:href="http://geoserver/wcs"
        method="POST">
        <wps:Body>
          <wcs:GetCoverage service="WCS" version="1.1.1">
            <ows:Identifier>topp:bluemarble</ows:Identifier>
            <wcs:DomainSubset>
              <gml:BoundingBox
                crs="http://www.opengis.net/gml/srs/epsg.xml#4326">
                <ows:LowerCorner>-180.0 -90.0</ows:LowerCorner>
                <ows:UpperCorner>180.0 90.0</ows:UpperCorner>
              </gml:BoundingBox>
            </wcs:DomainSubset>
            <wcs:Output format="image/tiff" />
          </wcs:GetCoverage>
        </wps:Body>
      </wps:Reference>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>cropShape</ows:Identifier>
      <wps:Data>
        <wps:ComplexData mimeType="application/wkt">
        <![CDATA[POLYGON((-19 48, 20 65, 46 48, 20 30, -19 48))]]>
        </wps:ComplexData>
      </wps:Data>
    </wps:Input>
  </wps:DataInputs>
  <wps:ResponseForm>
    <wps:RawDataOutput mimeType="image/tiff">
      <ows:Identifier>result</ows:Identifier>
    </wps:RawDataOutput>
  </wps:ResponseForm>
</wps:Execute>
```

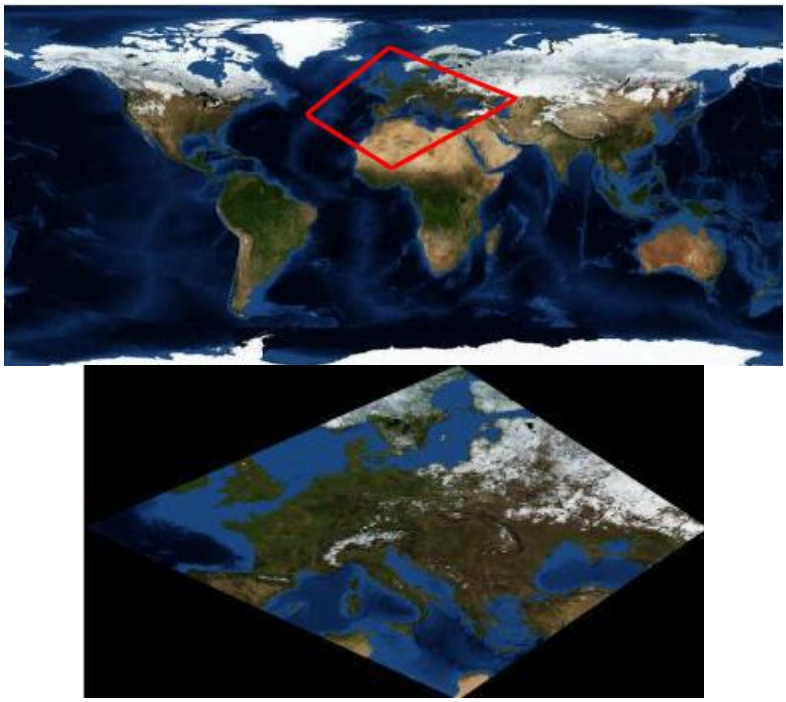

Figure 2.6: Crop Process Results

2.4 Comparative Study

In this section, the comparative study for raster calculation using the WPS implementations is provided for processing large data volumes. The specification of the system which was used to carry out the processing was as the following: CPU: Xenon 2.93 GHz, RAM: 12 GB and OS: windows 7 64bit.

The test data consisted of 2 Landsat scenes from Canada. The size and dimension of a normal Landsat scene is 145 MB and it has almost 8500*6000 pixels. The size of the scenes were reduced to 3MB having a dimension of 1500*1041 pixels (Figure 2.6).

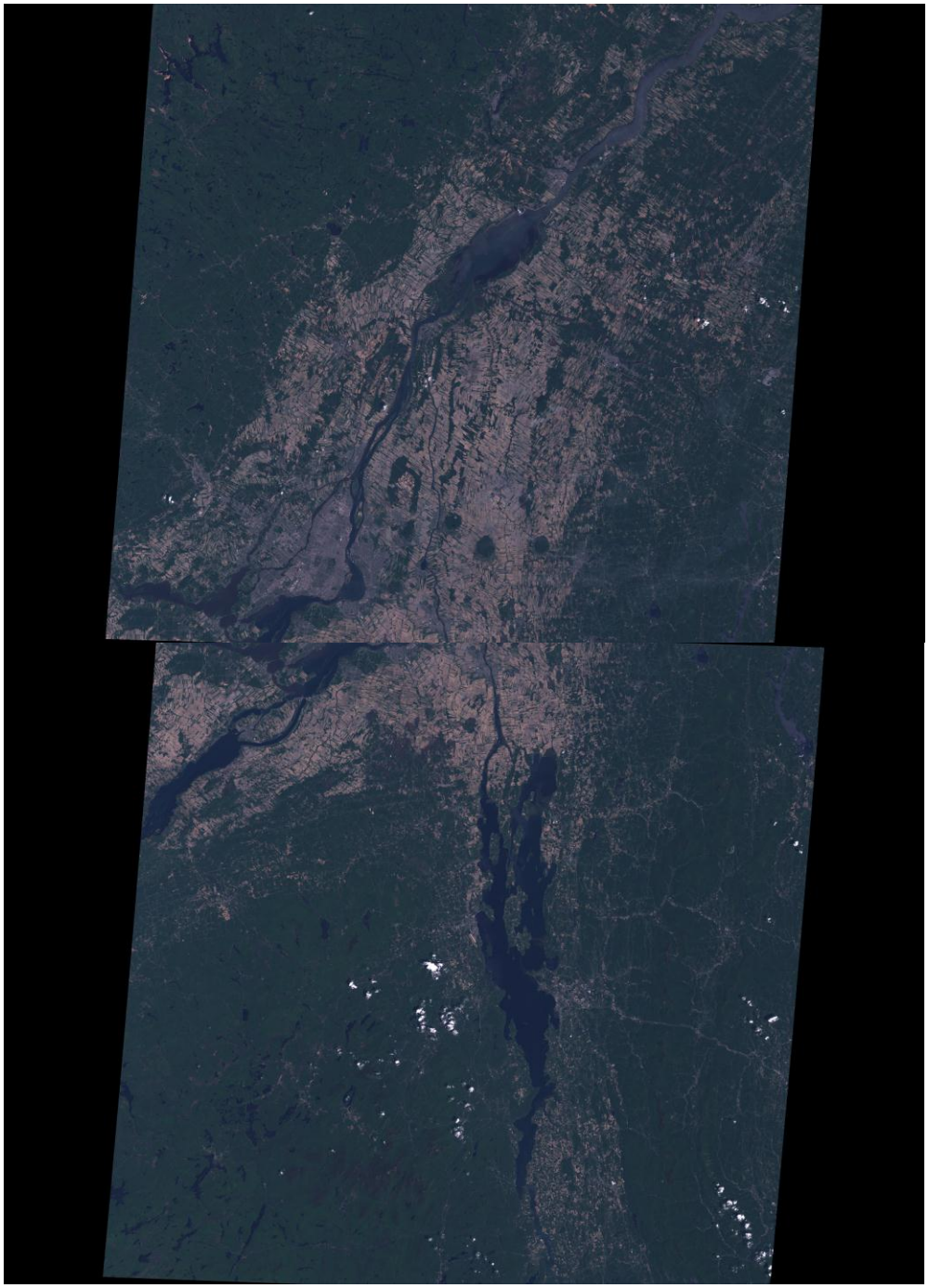

Figure2.7: Landsat Images for the Study Area

The algorithm used for the comparison is image multiplication. The algorithm takes the Digital Number (DN) value of each pixel in each band (red, green, blue) from one

image and multiplies it by the Digital Number of the corresponding pixel on the other image, and then puts the final value in a new pixel of a new image. The two images should have same number of rows and columns and pixels and must have the same extent.

Geoserver has this process in-built and as a result it could be run immediately, but for 52North and Degree and Zoo the code was rewritten in Java language and for PyWPS the code was written in Python language,

The pseudo code of this algorithm is as follows:

```
for i=1:length(image1(1,:,1))
  for j=1:length(image1(:,1,1))
    result_image(i,j,1) = image1(i,j,1) * image2(i,j,1);
    result_image(i,j,2) = image1(i,j,2) * image2(i,j,2);
    result_image(i,j,3) = image1(i,j,3) * image2(i,j,3);
  end
end
```

In order to make the test more realistic Map Server for Windows (MS4W) was installed on another computer (server) and the Landsat scenes were uploaded to the server. As a result the image multiplication was done by calling the images from another server and not the same computer on which the processing was being done. This had an effect on the time of the processing as network speed affects the whole process. In fact was necessary to have the Landsat scenes on another server because WPS is for doing calculations over networks and Internet. Also to reduce the processing time, the final image (the result of the process) was stored on the same machine where the process was

carried out. The process was carried out three times for each WPS implementation and the results listed in table 2.1 are the average of the processing times achieved:

Table2.1: Results of Implementation

Implementation	**Processing Time**
Geoserver	12’34”
52 North	13’12”
Zoo	13’38”
Deegree	13’56”
PyWPS	16’14”

It is noteworthy that the algorithm employed (image multiplication) in this case study is a default algorithm for Geoserver. This makes Geoserver run the processing faster in comparison with other WPS implementations. But another important reason that makes Geoserver the better implementation of WPS for raster processing is the existence of an advanced raster processing toolkit. This toolkit includes JAITools, ImageIO-Ext and GeoTools extensions, each one of these extensions add a unique process and property to Geoserver and make Geoserver more suitable to perform raster processing with.

For example JAITools extends, complements and replaces Oracle Java Advanced Imaging (JAI) Library. Using this library, advanced raster processing such as range look up, converting raster to vector, contour building, raster classification, raster statistics, raster algebra, color map extraction, polygon extraction and many more can be carried out using Geoserver (Jaitools, 2013).

Image IO – Extension from the other hand supports a wide range of image formats, from BigTiff, netCDF, MatFile5, HDF to JP2000. Also when integrated by GDAL other formats such as JPEG2000, MrSID, ECW, ERDAS Image, HDF4, Envisat, Arc/Info Grid will be supported.

GeoTools is an open source Java code library which provides standards compliant methods for the manipulation of geospatial data; for example, to implement Geographic Information Systems (GIS). GeoTool's raster process plugin adds extra features to raster calculation in Geoserver, processes such as creating image mosaic and image pyramids, on the fly color mosaicking, on the fly tiling, support for footprints on the tiles and many other advanced raster processes which can be done using GeoTools library (Geotools, 2013)

2.5 Conclusions

To Implement a WPS, regardless of the platform of it, there some points which need to be considered. The WPS protocol describes a mechanism by which a client computer may submit a job to be processed on a server computer, using uploaded data or data provided via a WFS or WCS service. This is classic client/server architecture, meaning that both a client component and a server component are needed.

For implementation and testing purposes it is useful to build the client-side component on a Geographic Information System (GIS) to take advantage of existing visualization features. However, initial testing may be performed with command-line or spatially unaware tools. The client-side component is the portion which handles XML

communication through the internet with the server, ideally without the user needing to directly see or work with the XML to discover available processes or to make their request and retrieve results.

The WPS server component could be implemented as a PHP web page, as an ASP.NET web page, as a standalone application, or implemented using any other server technology. Many of the operations needed by a WPS server are essentially metadata operations: providing information about individual processes (i.e., required inputs) and listing processes available on a server. The WPS server will ideally load information on available processes from a configuration file or from a database, thus making the code written for the WPS server reusable by adding additional processes to configuration files or to the database. These configuration files or this database may also indicate to the WPS server how to launch the process and how to parse its output files.

The following also should also be considered when planning to use WPS:

- WPS is particularly a good approach for lighter processes. However, for more complex and complicated processes, the process might take a long time to end. The processing time in WPS is a function of network speed, computing power, client/server configuration, input data and the algorithm itself. As each of these factors get more complex, for instance more servers involved or large data as an input to be processed, the chances of occurrence of a longer processing time will get higher. As a result WPS might not be suitable for real-time applications.
- When reading input files from different separate servers, WPS might not be the best solution as servers are not connected to each other, and if one goes down the whole process will be terminated. The main reason for this is because WPS is not

a distributed and scalable service by definition. This means that normally WPS processes are carried out on servers that are not interconnected to each other. Although there are efforts being done on providing distributed WPS services but there has not been a fully scalable distributed WPS server capable of handling large data processing yet.

- Servers have might have different loading and through traffic limitations. As a result, when reading inputs from a different server, the process can be terminated or can be taken a long time to proceed due to the problems from server side.
- Although WPS is set out to eliminate the need of end software for performing a process, it is always better for more complex operations to have a desktop GIS paired with the WPS platform in order to make the processing time faster and to be able to visualize the results better.
- Although all of the implementations of WPS follow OGC's standards, but custom coding algorithms and processes differ from each other for different platforms, as a result, it makes it difficult to switch between platforms if not satisfied with one.
- Even simple raster calculation, such as multiplication, proves to be time consuming when done by WPS in comparison with desktop GIS.

References

Brunner, D., Lemoine, G., Thoorens, F.-X. and Bruzzone, L. (2009), 'Distributed Geospatial Data Processing Functionality to Support Collaborative and Rapid Emergency Response', IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, Vol. 2, No. 1, pp. 33-46.

Deegree Documentation (2012). http://www.deegree.org/ Accessed 22 December 2012

Friis-Christensen, A., Lucchi, R., Lutz, M. and Ostlander, N. (2009), 'Service chaining architectures for applications implementing distributed geographic information processing', International Journal of Geographical Information Science, Vol. 23, No. 5, pp. 561-580.

Gebbert (2009). GRASS GIS wiki WPS section, http://grass.osgeo.org/wiki/WPS

Geoprocessing 52 north (2012), http://52north.org/communities/geoprocessing/ Accessed 11 December 2012

Geoserver Documents (2013), http://docs.geoserver.org/ Accessed 8 February 2013

Opengeo Documents (2013), http://suite.opengeo.org/opengeo-docs/processing/intro.html/ Accessed 9 January 2013

Geotools (2012).http://www.geotools.org/ Accessed 2 December 2012

Geoserver (2012). http://geoserver.org/display/GEOS/Welcome, Accessed 27 November 2012

Jaitools (2012).http://jaitools.org/ Accessed 2 December 2012

Li, X., Di, L., Han, W., Zhao, P. and Dadi, U. (2010), 'Sharing geoscience algorithms in a Web service oriented environment (GRASS GIS example)', Computers & Geosciences, Vol. 36, No. 8, pp. 1060- 1068.

Lowe, D., Woolf, A., Lawrence, B. and Pascoe, S. (2009), 'Integrating the Climate Science Modelling Language with geospatial software and services', International Journal of Digital Earth, Vol. 2, No. s1, pp. 29-39.

Michaelis, C. and Ames, D.P., (2008), 'Considerations for Implementing OGC WMS and WFS Specifications in a Desktop GIS'. Journal of Geographic Information System, Vol 4 No.2., 161-167

Open geospatial (2012).http://www.opengeospatial.org/ Accessed 11 December 2012

Open Geospatial Consortium Inc., (2005), 'OpenGIS® Web Map Services', Ref number: OGC 05-007r4, Version: 0.4

Open Geospatial Consortium Inc., (2009), 'OpenGIS® Web Map Services - Profile for EO Products' Ref number: OGC 07-063r1, Version: 0.3.3

PyWPS (2013). http://wiki.rsg.pml.ac.uk/pywps/PyWPS, Accessed 16 January 2013

ZOO Project (2012). http://zooproject.org/site/ZooWebSite/Demo/SpatialTools#ZOOspatialtoolsdemo/ Accessed 13 December 2012

52North (2012).http://52north.org/ Accessed 23 December 2012

Chapter 3

3 A Comprehensive Overview of Cloud Computing in GIS

Abstract

Cloud computing is an Internet-based supercomputing principle, practice and technology, which supplies dynamic, scalable, and pay-per-use services and it potentially has extensive computing and storage capacity with high reliability and security. By utilizing cloud computing, users can have the advantage of sharing spatial data and applications, using specific GIS services designed for their needs with high and scalable hardware. The aim of this article is to review the current limitations and benefits, of cloud computing that can make an impact on the GIS field, mainly focussing on the perspective of storing, analysing and presenting geospatial data. Such a review is important since it helps in better understanding of GIS in respect with cloud computing. First the concept of cloud computing, its general architecture and applications are described, definitions of cloud computing are provided and based on these definitions a new definition is proposed in order to discuss the role of cloud computing in GIS. Finally, some suggestions are made for the future of cloud computing in GIS.

3.1 Introduction

Cloud Computing is a "large-scale distributed computing paradigm that is driven by economies of scale, in which a pool of abstracted, virtualized, dynamically-scalable, managed computing power, storage, platforms, and services are delivered on demand to external customers over the Internet (Foster et al., 2008). In the early 2000s, network computing and Internet computing evolved to grid computing in order to obtain the computing power on demand due to the increase in size of data and need in specific services (Figure 3.1). (Foster et al., 2008) stated, “cloud computing has indeed evolved out of Grid Computing and relies on Grid Computing as its backbone and infrastructure support”. The evolution has been the result of shifting from an infrastructure that delivers storage and compute resources to a platform that delivers more abstract resources and services such as Netflix or Salesforce in which the user only deals with a rather simple interface to get his service and the complex computation and hardware resources are not among the concerns of the user.

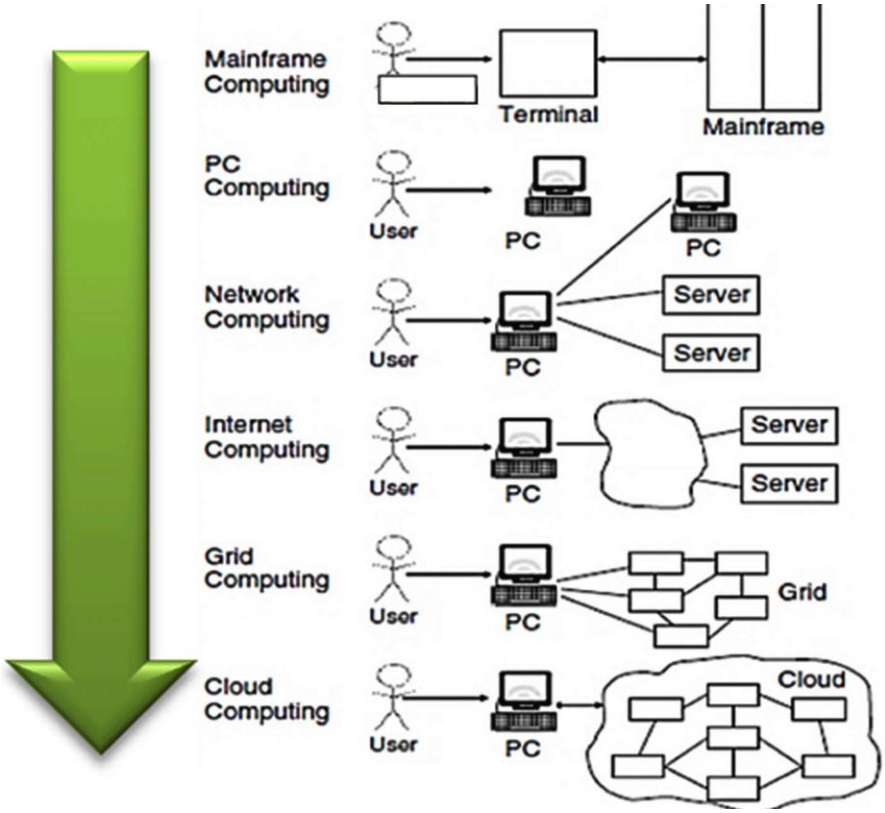

Figure 3.1. Evolution of Computing Platforms (Hand book of cloud, 2010)

This computing evolution has been also observed in other fields. Today GIS is moving towards providing abstract services to meet the needs of its users rather than staying in the form of a general product such as ArcGIS software where a whole general package of tools are available for a user who might not need everything in that package. The federal, state and local governments as well as the private sector, as the main GIS users, are slowly but steadily moving towards the adoption of cloud to improve handling and sharing of their data resources and promote collaboration among them. Examples can be found in the governments of USA, Canada, UK, Japan, Australia and South Korea, that have already defined their cloud computing strategy and are determined to

run government clouds and design data centers in the cloud, leveraging public clouds where appropriate (Oracle iGovernment white paper, 2012).

This Chapter aims to describe the benefits and limitations of cloud computing in the GIS field. In order to achieve this goal, the first section explains different definitions of cloud computing using their common grounds and divergences. A definition of cloud computing is given to further explain the meaning and concept of cloud computing Moreover, cloud computing is discussed in terms of technology, delivery model, and a rising framework. The architecture of cloud computing is then investigated and compared to traditional architectures of delivering web applications such as SOA.

In Section 2, the focus is on the role of cloud computing in the GIS field and its expected impact on the GIS community. Therefore, cloud computing is discussed from the functionality perspective of storing, analyzing and presenting geospatial data. Towards this end, a review on technologies enabling these functionalities is provided.

In section 3, applications of cloud computing in GIS are briefly presented as well as the rising trends, main advantages and disadvantages of cloud computing. The Chapter concludes by discussing the ways GIS can benefit from cloud computing.

3.2 Definition of Cloud Computing

A variety of definitions of cloud computing can be found in the literature and there is little consensus on the definition of cloud. However, the definition of cloud computing from a GIS perspective is important to determine the barriers and facilitators of using

cloud computing in GIS. Therefore, it is important to first reach a comprehensive definition of cloud computing.

The first definitions of cloud computing date back to 2007 and they were often mixed with the concepts of grid computing and Internet computing mainly because the concept of cloud computing was just beginning to form, and many researchers then considered that cloud computing was grid computing or even Internet computing. Sarathy et al., (2010) describe "Cloud Computing, to put it simply, means "Internet computing." The Internet has been commonly visualized as clouds; hence the term "cloud computing" for computation done through the Internet" seemed to be the correct one. Weiss (2007) has also defined it as "a way of computing which shares computer resources on Internet instead of using software or storage on a local computer".

Other definitions of cloud computing from the perspective of big data, have been used interchangeably with the concept of Data Centers. For example, Singh and Gupta (2011) describe cloud computing associated with a new paradigm for the provision of a computing infrastructure. This paradigm shifts the location of infrastructure from desktop to the network to reduce the costs in management of hardware and software resources.

However, it is important to point out that cloud computing has many features in common with grid computing and Internet computing such as they both help to reduce costs of computing, they have excessive hardware and processing power, they are scalable, reliable and flexible and they enable sharing resources over Internet. However, one of the main differences of cloud computing with the previous computing paradigms, is that cloud computing is service based, and being service based shows itself in this

definition of cloud by Lu (2010) who states that Cloud Computing an approach to computing in which dynamically scalable computing hardware and software resources are provided as a service over the Internet Wang and Lazewski's (2008) define cloud computing as a set of network enabled services, providing scalable, Quality of Service (QoS). QoS guaranteed, normally personalized, inexpensive computing platforms on demand, which could be accessed in a simple and pervasive way. Furthermore, Armbrust et al., (2009) refer to both the applications delivered as services over the Internet and the hardware and systems software in the Data Centers that provide those services. Moreover, Mortier et al. (2008) argue that cloud computing allows the access to files, data, programs and 3rd party services from a Web browser via the Internet that are hosted by a 3rd party provider and by “paying only for the computing resources and services used.” The services themselves have long been referred to as Software as a Service (SaaS).

Among other features of cloud computing, virtualization also plays an important role as Jing and Zhang (2010) defined cloud computing as “a novel architecture to eliminate massive system requirements from end user to system administrator”. Its best feature is flexibility. Virtualization technology provides this flexibility in cloud computing since heavily loaded server tasks can be easily migrated to light-loaded servers. In a normal case, a server can serve one purpose at a time, but using virtualization, multiple virtual servers can be created on one single actual server and as a result, the defined tasks will be divided among the virtual servers.

Another definition of cloud computing is the one proposed by National Institute of Standards and Technology in 2013 (NIST, 2013) states that "Cloud computing is a

model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage applications and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. A similar definition is also given by Kang (2011) in which cloud is a network shared scalable resource pool that can provide services such as computing power and/or storage space in abstracted and encapsulated forms, which are delivered through networks with proper levels of isolation, dynamic resource allocation and usage monitoring, to authorized users based on users' demands.

Considering the above definitions, the key characteristics of cloud computing can be summarised as being on-demand self-service, broad network access, resource pooling, rapid elasticity, measured service, virtualization and scalability to provide comprehensive as well as on-demand services. Based on these characteristics, this thesis proposes a definition which contains all the key characteristics. The definition is the following:

- Cloud computing is a scalable framework consisting of a set of connected and virtualized computers over the Internet, for enabling on-demand network access to a shared pool of configurable computing resources as a service.

As a new paradigm, it is expected that cloud computing will continue to be discussed from different perspectives. After reviewing the definitions of cloud computing, cloud computing will be looked at from five different perspectives in the next section. which are: a new computing paradigm, a new framework, a computing architecture, a service and a deployment model.

3.2.1 Cloud Computing as a New Computing Paradigm

Figure 3.1 shows the evolution of computing, the shift from grid computing to cloud computing that has occurred in 2007. It can be argued that cloud computing is a change in the delivery model rather than a technology shift, since the backbone of the cloud is grid computing.

Distributed computing, as the basis of grid computing, is not a new concept, accessing large computing and processing power has long been a goal for computer scientists and it was in 1978 that. Enslow defined distributed data processing systems and a new paradigm in computing was born (Enslow, 1978). Since then technology and resources have changed and increased dramatically but the concept has remained the same that a distributed system is collection of independent computers or machines that are connected to each other to provide more processing power and appear to its users as a single coherent system.

This definition is the backbone of both grid computing and cloud computing. In mid 1990s the term grid computing started to be used as technologies that would allow consumers to obtain computing power on demand. On demand was added to the definition because Internet was already becoming the main framework and was able to bring computing power on demand. In 2007 cloud computing was introduced as a large-scale distributed computing paradigm in which a pool of abstracted, virtualized, dynamically-scalable, managed computing power, storage, platforms, and services are delivered on demand to external customers over the Internet. Cloud computing was started to be used as a paradigm more than just computing power on demand, but as an

abstract, virtual, service based delivery model with the computing power and hardware power of grid computing.

3.2.2 Cloud Computing as a New Framework

The paradigm shift of computing has also led to a change in the framework in which that paradigm should operate. Cloud computing was the result of the rise of data centers which were demanding for a large volumes of data to be accessed at a very high speed. New scalable frameworks were needed to be able to handle and analyse this big data, mainly, to handle virtualization and abstraction which could automatically load balance among the computers in a network and to automatically add and remove nodes from this network. And more importantly, a framework was needed with the ability to handle all types of data, i.e. un-structured, semi-structured and structured.

It was these requirements that resulted in the development of new framework such as the Apache Hadoop Project (Hadoop, 2005) that allows for the distributed processing of large data sets across clusters of computers. It implements a computational framework named as MapReduce which supports the distributed processing of large data sets, where an application is divided into many small fragments of work, each of which may be executed or re-executed on any node in the cluster. Hadoop is currently the *de facto* framework for cloud computing, being used by the largest cloud owners and providers (e.g. Yahoo!, LinkedIn, Facebook, and Google) to store and analyze any type of data.

3.2.3 Cloud Computing as a New Computing Architecture

Cloud computing as a new architecture offers a different service delivery, a personalized service based on users' preferences from a pool of shared resources. On the other hand, service based architectures are not new concepts. SOA and Web 2.0 along with web technologies such as web mashups and web applications have been used to deliver services prior to the cloud computing era (Schroth and Janner, 2007)

Service Oriented Architecture (SOA) is a flexible set of design principles used during the phases of systems' development and integration. It is an architecture to integrate silo applications and the functionalities are accessed as services over the Internet. Web 2.0 was introduced by O'Reily (2006) to transit web sites from isolated information centers to interlinked computing platforms. Data was the driving force of the architecture and the main features of the architecture were user participation and user content production. Some examples include Wikis and social networks. Web mashups are web applications that combine data from more than one source and integrate them in one platform. News mashups, video and photo mashups and mapping mashups are some examples of it. In particular, mapping mashups have been used in a variety of applications in the GIS field (Li and Gong, 2008).

The question is why cloud computing is a new architecture in comparison to the existing architecture? The answer lies on the components of the cloud. Foster et al., (2008) define four components: the Fabric, the Unified resource, the Platform, and the Application (Figure 3.2).

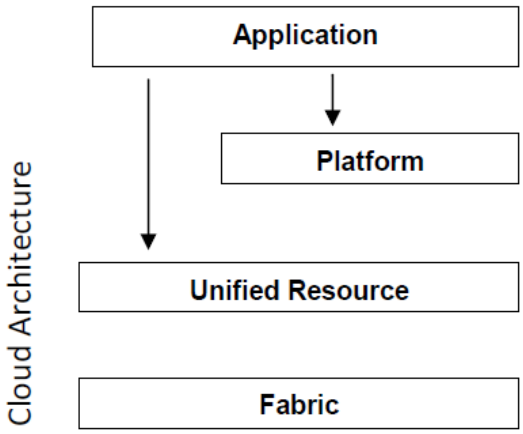

Figure3.2 Cloud Computing Architecture

The fabric component contains the raw hardware level resources, such as compute resources, storage resources, and network resources. The unified resource layer contains resources that have been abstracted, usually by virtualization so that they can be exposed to upper layer and end users as integrated resources, a virtual computer or cluster for instance. The platform component adds on a collection of specialized tools, middleware and services on top of the unified resources to provide a development, for instance, a Web hosting environment. Finally, the application component contains the applications that would run in the cloud. Every cloud architecture contains all four levels (Tianfield 2011), from infrastructure to interface of a service. In contrast, the existing architectures such as SOA only supports the delivering of the service without supporting the infrastructure needed for delivering such a service.

3.2.4 Cloud Computing as Services

In general, Cloud Computing services fall into one of the following categories (Handbook of cloud, 2010):

- Infrastructure as a Service (IaaS): provides computing infrastructure such as encapsulated virtual servers, virtual desktops, storage units and network resources and they are delivered as services over networks. One example is Amazon's Elastic Cloud Computing (EC2) Service.
- Platform as a Service (PaaS): provides an API for end-users so they can deploy applications and/or services based on cloud provider platforms. A typical example is Google App Engine that can be used to host or develop web applications.
- Software as a Service (SaaS): delivers software over networks to end-users. Since this type of service provides features of specific software, SaaS usually only provides functionalities to satisfy specific needs. Salesforce is an example of a SaaS provider.

SOA and Web 2.0 have a similar architecture to SaaS, but SaaS makes it possible for a software to be deployed as a hosted service and be accessed over the Internet. Moreover, it enables access to commercially available software by providing the grounds for subscription based software licence and pay as you go license. Table 3.1 summarizes the existing cloud computing services.

Table 3.1: Summary of Cloud Computing Services

	Software as a Service SaaS	Platform as a Service PaaS	Infrastructure as a service IaaS
Applications	Communications (Email) Productivity tools (office) Website testing Virtual Desktop	Database management Application Development and Deployment Service Test	Servers Storage Management Network Security
Cloud Providers	Netsuite Oracle Salesforce.com	Google App engine Rack Space cloud site Force.com	Google Microsoft Azure Amazon web services
Users	End Users	Application Developers	IT operators, Network architects

3.2.5 Cloud Computing as a Deployment Model

The deployment models of cloud computing are public cloud, private cloud, hybrid cloud and community cloud.

- Public cloud: It is the traditional mainstream cloud deployment technique whereby resources are dynamically provisioned by third party providers who share them with the users. In a public cloud model, services are created, maintained and delivered as software or platform (SaaS or PaaS) by a cloud vendor. Users execute all applications and store all data in a cloud provider's infrastructure (Figure 3.3). A cloud user may still have his private data stored on the cloud server which will only be accessible to certain people defined by him. If any cloud user wants to process or share any data, he will have to upload the data to the cloud server (Handbook of cloud, 2010).

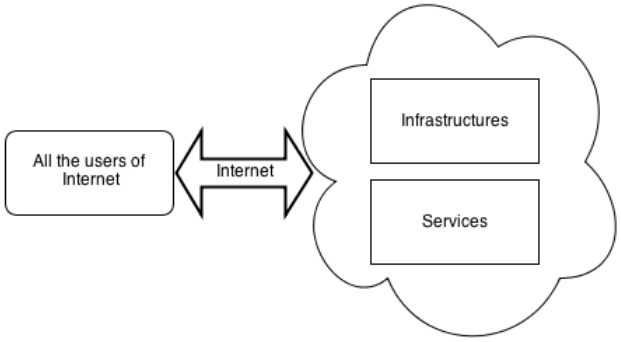

Figure 3.3: Public Cloud Model

- Private cloud: In this cloud deployment technique, the computing infrastructure is solely dedicated to a particular organization, business or individual. It is a more secure deployment because it belongs exclusively to a particular organization. A private cloud is built, operated and maintained by the owner. The hardware of a private cloud can be an in-house-cloud or a virtual-cluster of a third party cloud computing providers' infrastructure. In either case, the user needs to create the cloud servers, deploy and distribute applications in the cloud, acquire and update data and maintain hardware, software and data (Figure 3.4) (Handbook of cloud, 2010).

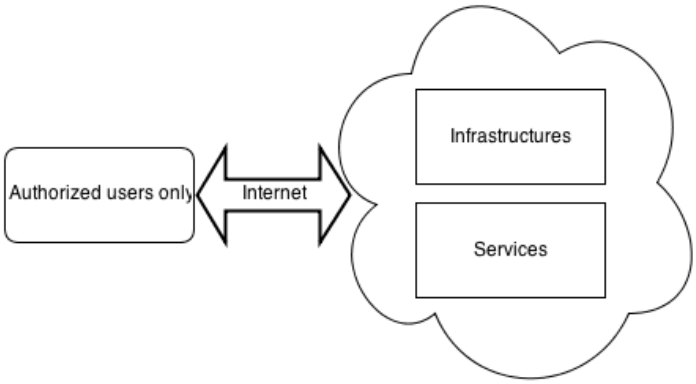

Figure 3.4: Private Cloud Model

- Hybrid cloud: This deployment technique integrates the positive attributes of both the public cloud and the private cloud models. For instance, in a hybrid cloud deployment a reasonable design might be to host private or highly-confidential data in a private cloud system and to host other data and applications in a public cloud system (Figure 3.5) (Handbook of cloud, 2010).

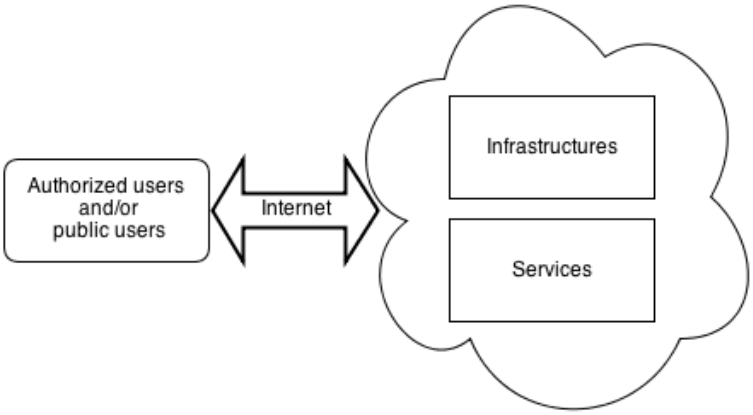

Figure 3.5: Hybrid Cloud Model

- Community cloud: This deployment technique is similar to a public cloud with the only difference being the distribution of the sharing rights on the computing resources. In a community cloud, the computing resources are shared amongst organizations within the same community. Therefore, this cloud covers a particular group of organizations, which share same interests and have similar jobs to be done on a cloud and need similar functionalities from a cloud.

In Table 3.2, a summary of the cloud deployment models is provided.

Table 3.2: Summary of cloud deployment models

Cloud Type	Data	Reliability, Availability	Privacy	Performance	Sharing ability	Complexity
Public	Third party	Good	Weak	Good	Easy	high
Private	In house	Poor	Strong	Fair	Fair	medium
Hybrid	Both	Very Good	Strong	Depends	Easy	low

3.3 Storing Data in Cloud Computing

The first step in every information system is to be able to store and save the data which has been obtained or collected or provided for the system. Cloud storage has several advantages over traditional data storage. For example, if the data is stored on a cloud storage system, that data can be accessed from any location that has Internet access. Cloud based storage systems allow other people to access the data, turning a personal project into a collaborative effort. This helps GIS in creating and sharing maps, the size of maps can often be huge and having the ability to share your map online with the resources for everyone to collaborate is of value for GIS.

An example for this is Google Docs (Google docs, 2007) which allows users to upload documents, spreadsheets and presentations to Google's data servers. Users can edit files using a Google application. Users can also publish documents so that other people can read them or make edits. Other examples are Websites like Flickr (Flickr, 2004) and Picasa (Picasa, 2004) which host millions of digital photographs. Their users

create online photo albums by uploading pictures directly to the services' servers and their cloud. There are also commercial platforms for storing data, Azure (Microsoft, Windows Azure, 2009) is a large-scale distributed storage system. Amazon web services (Amazon Web Services, 2010) also provides computing and data storage infrastructure which can be a host for any type of data, as well as applications.

The two biggest concerns about cloud storage are reliability and security (You et al., 2012)**.** Clients are not likely to entrust their data to another company without a guarantee that they will be able to access their information whenever they want and no one else will be able to get at it. Cloud storage systems generally rely on hundreds of data servers. Since computers occasionally require maintenance or repair, it is important to store the same information on multiple machines. This is called redundancy. Without redundancy, a cloud storage system cannot ensure clients that they have access their information at any given time.

To secure data, most systems use a combination of techniques, including: Encryption which means they use a complex algorithm to encode information, Authentication processes, which require creating a user name and password. Authorization processes the client lists the people who are authorized to access information stored on the cloud system.

3.3.1 Manage and Process Data in Cloud Computing

Data management in cloud computing differs from prior approaches in several aspects (Gannon et al., 2011), First, cloud computing consolidates computing capabilities and

data storage in very large datacenters or local datacenters that need to be more careful with their expenses. Second, hosting data in a centralized facility can be a catalyst for data sharing across different scientific domains. Finally, from the perspective of data hosting, cloud computing is inherently more sustainable because the data is stored on different servers and potentially in different physical locations with the servers connected to each other. If one or multiple servers fail, the remaining servers will automatically back up all the data and the whole cloud will continue to operate.

To be able to manage large volume data in the cloud, scalable and distributed databases and file systems are needed. Below are some of the most used scalable databases and file systems. The MapReduce (Dean and Ghemawat, 2004), proposed by Google in 2004 is a framework and programming model for data-intensive distributed computing that enables processing of a high-volume dataset on a large number of machines and its implementation by Apache, called Hadoop is being used by Google, Microsoft, Yahoo and many other cloud providers.

The Google File System (GFS) (Ghemawat, Gobioff, & Leung, 2003) fits in with the Map-Reduce programming model. GFS disperses a large file onto a set of machines each with its own commodity hard drives. BigTable (Chang et al., 2006) also developed by Google is another distributed storage system for managing just structured data that is designed to scale to a very large size. Several other open source projects such as HBase (HBase, 2010), Hypertable (Hypertable, 2010) and Hive (Hive, 2013) are available for cloud based analysis as well. HBase and Hypertable make use of Hadoop to implement a distributed store that mirrors the software design of Google's BigTable. Apache's Hive,

a derivative project of Hadoop, is a data warehouse infrastructure, which allows SQL-like ad hoc querying of data stored in Hadoop's file system.

Cassandra (Lakshman and Malik, 2010) developed by Facebook and Yahoo is a distributed storage system for managing very large amounts of structured data spread out across many commodity servers.

There are many other projects which have dealt with providing distributed storage or higher-level services over wide area networks. Spatial data, as data with location information, is increasing in volume and size and thus, should be managed and processed using distributed and scalable frameworks. There have been a number of projects addressing spatial data analysis in scalable and distributed frameworks, ESRI has developed a toolkit called GIS tools for Hadoop which provides spatial framework and geoprocessing tools for Hadoop. This toolkit enables spatial operations and queries to be done on spatial data stored in Hadoop and also facilitates visualization of spatial data in ESRI's platform.

Open source projects such as Geomajas (Geomajas, 2013) which is a scalable server side GIS framework allowing multiple users to control spatial data with their browsers or Paradise, which is a parallel database system for GIS applications have also addressed issue of big data analytics and also cloud computing in GIS.

3.3.2 Visualizing Data in Cloud Computing

Data visualization is the study of the visual representation of data, meaning information that has been abstracted in schematic form, including attributes or variables

for the units of information. There are a variety of conventional ways to visualize data – tables, histograms, pie charts, bar graphs and standard maps are being used every day, in almost every project. However, to convey a message to users effectively, sometimes more effective tools are needed, from the other hand and with huge growth in big data, structured and unstructured, the need for abstraction and visualization of the data to make better decisions has made visualization an important part of every information system. As Forrester Reserach (2010) states "Widespread adoption of mobile technology and social computing has driven interest in visualization capabilities and real-time analytics"

Looking at the trends of data visualization, especially in the recent years, the role of cloud computing can be seen in further enhancing the tools for data visualization, trends such as growth in data volume, 3D visualization, Augmented Reality (AR) and mobile visualization. With growth in data volume need for advanced data visualization Forrester Research (2010) states "Now, through advanced data visualization, potential exists for nontraditional and more visually rich approaches, especially in regard to more complex (i.e., thousands of dimensions or attributes) or larger (i.e., billions of rows) data sets, to reveal insights not possible through conventional means"

Features of advanced data visualization are: dynamic data interaction, linked multi-dimensional visualization and personalization, cloud computing can be a suitable platform to help in this case., One of the most prominent features of cloud computing is to provide personalized service to particular users and a key benefit of advanced visualisation tools is that the approach to project delivery can be changed using these tools.

3.4 Cloud Computing Advantages and Disadvantages

3.4.1 Advantages of Cloud Computing

The main advantages of cloud computing are:

- **Scalability**: Users are able to scale resources allocated for applications up or down dynamically and flexibly, rather than acquiring additional infrastructure such as hardware and software to support applications when demand is high. Users can cut the lead time by quickly scaling up in a cloud environment while avoiding the risk of idling the infrastructure when demand is low. This advantage is probably the most important advantage from GIS perspective as the size and volume of geospatial data are increasing dramatically and there is a need for scalable storage and analytical power.
- **Virtualization**: Deploying, managing and maintaining applications can be one of the most costly and time-consuming aspects of client computing. Virtualization allows consolidating and running applications onto fewer physical servers, which drives up server utilization rates. Additionally, virtualization enables quick provisioning and deployment, improved workload balancing, and enhanced resiliency and availability by giving the ability to dynamically move VMs from one server to another.
- **Multi-tenancy:** Resources in cloud frameworks can be shared among a large number of users and applications. Every application in the cloud needs its own

secure and exclusive virtual computing environment and this environment can encompass all or some select layers of the host cloud's architecture. The cloud computing's framework and platform make multi tenancy possible by virtualization.

- **Shared resource pool:** Different data types with various sizes can be handled and shared easily using cloud without the fear of hardware or software shortage. A while ago, the most popular way to share or distribute map data through the internet was using ArcIMS. Today, many new tools such as MapServer and GeoServer have emerged to share data through an Intranet or the Internet. But none of them can provide the massive network throughput like a cloud infrastructure. The Cloud infrastructure is pushing the ease of sharing and distributing data further because of its scalable distributed network-based nature.
- **More efficient application/data update model**: In most Desktop systems and desktop GIS, software updating is not an easy task. Update patches have to be created, distributed and installed by end users or IT administrators. In a cloud platform, users will not even notice the update of applications since everything is done on the cloud server side.
- **Pay-per-use:** The users pay most likely much less for the services, because they pay only for the computing resources and services they use, and the subscription-based or pay per- use charges are likely much lower than the cost of maintaining on-premises computing resources.

3.4.2 Disadvantages of Cloud Computing

The main disadvantages of cloud computing are:

- **Privacy and data security**: the issue of privacy and data security has always been a major concern for users who had data stored in a third-party storage space (You et al., 2012). In cloud computing, this issue is also the most important concern of cloud users as they will not know exactly where their data are going to be stored and who would have the privilege to access it inside the cloud infrastructure and how the safety and integrity of their data can be guaranteed.
- **Data lock-in:** certain cloud providers tend to provide a lock-in data structure by providing non transparent exclusive data storage methods to users. This may cause serious inconveniences in GIS systems because of the fact that GIS users of a cloud cannot choose to change the cloud provider if the infrastructure of the cloud is not compatible with their GIS system. This means that if, for instance, the cloud does not support a particular geographical data type, then the user has to handle all the costs of changing the provider.
- **Services Interoperability:** Currently, cloud computing doesn't have enough support for the interoperability of services. This brings several problems for cross-platform services or services between different services. As an example, a user needs to create a map with a service, run a number of processes on it with another service, and export the results on a third service for visualization, this cannot be done in one cloud service as different services can't operate together.

3.5 Applications of Cloud Computing in GIS

GIS, just like any other information system, deals with data. As a result, with rise of big data and fast growth rate of data, geospatial data has also been growing rapidly. This makes cloud computing a suitable solution for GIS. GIS applications can be moved to cloud just like non spatial applications did, Netflix and Kindle are examples of non-spatial cloud based applications that are working in cloud computing infrastructures..

Several cloud-based GIS solutions: ArcGIS Server on Amazon EC2, ArcGIS.com, ArcLogistics and Business Analyst Online (BAO). ArcLogistics and Business Analyst Online (BAO) are in the category of software as a service (SaaS) provided by ESRI. ArcLogistics can provide users with optimized routes and schedules based on multiple factors. BAO is a web-based service that provides reports and maps based on locations. The giscloud (giscloud, 2013) is a cloud based GIS system Users can upload, edit, convert, create and visualize GIS data through Internet browsers by using the services provided by the giscloud.

Urban observatory (Urban Observatory, 2013) is probably one of the well-known examples of GIS cloud based platforms. Urban Observatory is an online mapping tool, uses maps to compare data from 16 major world cities. Launched by ESRI and TED, the urban observatory lets users compare 35 various social and cultural aspects of each city. Users are able to select three cities and one theme at a time to compare from youth and senior population to commercial and industrial land usage and get answers to most common GIS questions regarding cities. Urban Observatory allows users to contrast the unique problems that exist in urban areas. Also users can contribute to the platform by uploading their own data.

Considering the growing applications of cloud computing in GIS and its advantages in this field, it can be said that more platforms and systems will adopt cloud computing in the near future and a larger number of GIS applications and services will be run in clouds.

3.6 Conclusions

Over the last decade, the cloud computing model has emerged as a solution for many problems in traditional IT infrastructure. It is expected that this technology brings an overwhelming capability of computing, fast microprocessor, huge memory, high speed network and reliable system architecture by solving the existing issues and challenges such as storing and processing big data, and a new era of future generation of computing through cloud computing technology will begin.

In this paper an overview on cloud computing's concepts, definitions and technologies have been provided, cloud's architecture and different delivery models have been explained and the impact of cloud on GIS was discussed. A new approach to cloud computing was also explained in which cloud computing was studied from the perspective of storing, processing and visualizing. Five application of cloud computing in GIS were reviewed. Advantages and disadvantages of cloud computing were discussed as well. But the future of cloud computing still remains rather unclear due to the very fast developments in this field. These developments include huge data centers which are adopting cloud computing technology more than before and mobile cloud

computing in which all the information, data, software and even hardware of users can be controlled through their mobile devices.

Developments in cloud based GIS applications and platforms will continue and more GIS services will be available in the clouds as cloud computing technologies continue to expand with a very high speed.

References

Armbrust, M., Fox, A., Griffith, R., Joseph, A., Katz, R., Konwinski, A., Lee G. and Stoica I., (2009), 'Above the Clouds: A Berkeley View of Cloud Computing,' Technical Report No. UCB/EECS-2009-28, University of California, Berkeley, 2009.

AWS (2010), Amazon web services, http://aws.amazon.com/ Accessed 23 September 2013

Business Analyst Online (2013). http://bao.esri.com/ Accessed 28 August 2013

Chang, F., Dean, J., Ghemawat, S., Hsieh, W. C., Wallach, D. A., Burrows, M. T., Fikes, A. and Gruber, R. E. (2006), 'Bigtable: a distributed storage system for structured data', In Proceedings of OSDI 2006, Seattle, WA

Dean, J. and Ghemawat, S. (2004), 'MapReduce: Simplified Data Processing on Large Clusters', Sixth Symposium on Operating System Design and Implementation (OSDI04), 2004

Enslow, P.H. (1978), 'What is a "Distributed" Data Processing System?' Computer Journal, Volume 11 Issue 1, January 1978 Pages 13-21

Foster, I., Vöckler, J., Wilde, M. and Zhao, Y. (2008), 'Cloud Computing and Grid Computing 360-Degree Compared', SSDBM 2008: 37-46.

Furht, B. and Escalante, A., (2010), 'Handbook of Cloud Computing', © Springer Science, Business Media, LLC 2010, ISBN 978-1-4419-6523-3

Flickr (2004), https://www.flickr.com/ Accessed 20 September 2013

Forrster Research Inc., (2010), 'BI Adoption Trends In Asia Pacific: High Priority, Poor Execution', available at: http://blogs.forrester.com/michael_barnes/12-05-17-bi_adoption_trends_in_asia_pacific_high_priority_poor_execution

Gannon, D., Barga, R. and D. Reed, (2011), 'The client and the cloud: Democratizing research computing', IEEE Internet Computing, vol. 15, no.1, pp.72-75

Ghemawat, S., Gobioff, H., and Leung S. (2003), 'The Google file system', SOSP'03, October19-22, 2003, Bolton Landing, New York, USA.

Geomajas (2013), http://www.geomajas.org/ Accessed 12 August 2013

Google docs (2007), http://docs.google.com/ Accessed 2 November 2013

GIS Cloud (2013). http://www.giscloud.com/ Accessed 1 September 2013

Hadoop (2005). http://hadoop.apache.org/ Accessed 12 September 2013

HBase Project (2010), http://wiki.apache.org/hadoop/Hbase/ Accessed 25 September 2013

Hive (2013). http://hive.apache.org/ Accessed 2 October 2013

Hypertable (2010), http://hypertable.com/why_hypertable/ Accessed 7 September 2013

Jing, X. and Jian-jun, Z. (2010), 'A Brief Survey on the Security Model of Cloud Computing', Ninth International Symposium on Distributed Computing and Applications to Business Engineering and Science (DCABES), 2010, pp.475-478

Kang, C. and Eastman, J.R. (2010), 'A cloud computing algorithm for the calculation of Euclidian distance for raster GIS'.

Lakshman A, Malik P. (2010), 'Cassandra: a decentralized structured storage system', SIGOPS Operating System Review , vol. 44, no. 2

Li, S., and Gong, J. (2008), 'Mashup: a New Way of Providing Web Mapping and GIS Services' ISPRS Congress Beijing 2008, Proceedings of Commission IV, 2008, pp. 639-649.

Lu, X. (2010), 'An Approach to Service and Cloud Computing Oriented Web GIS Application', 978-1-4244-5143-2/2010 IEEE, pp.1-4

Microsoft Windows Azure, Chappel, D., David Chappel & Associates, (2009), 'Introducing windows Azure', Microsoft, Inc, Tech. Rep.2009

Mortier, R., Madhavapeddy, A., Crowcroft, J. and Hand, S. (2010), 'Multiscale not multicore: efficient heterogeneous cloud computing', published by the British Informatics Society Ltd. Proceedings of ACM-BCS Visions of Computer Science 2010

National Institute of Standards and Technology document on cloud computing (2013), http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf

Oracle iGovernment White Paper, (2012), Oracle's Cloud Solutions for Public Sector, March 2012, available at http://www.oracle.com/us/industries/public-sector/cloud-solutions-public-sector-wp-323002.pdf

O'Reilly, T. 'What is Web 2.0', At: http://oreilly.com/pub/a/web2/archive/what-is-web-20.html?page=5/ Accessed 14 October 2013

Picasa (2004), http://picasa.google.ca/ Accessed 20 September 2013

Sarathy, V., Narayan, P. and Mikkilineni, R. (2010), 'Next Generation Cloud Computing Architecture: Enabling Real-Time Dynamism for Shared Distributed Physical Infrastructure', International Workshop on Enabling Technologies: Infrastructures for Collaborative Enterprises (WETICE), 2010 19th IEEE , June 2010, pp: 48-53

Singh, P. and Gupta, R. D. (2011), 'Development of Software As a Service Based GIS Cloud for Academic Institutes', Geospatial world forum, January 2011, available at: http://www.geospatialworldforum.org/2011/proceeding/pdf/pushprajfullpaper.pdf

Schroth, C. and Janner, T. (2007), Web 2.0 and SOA: Converging Concepts Enabling the Internet of Services. IEEE IT Professional Vol.9, No.3, pp.36-41

Tianfield, H. (2011), 'Cloud Computing Architectures'. Proceedings of 2011 IEEE International Conference on Systems, Man and Cybernetics (SMC' I I), Anchorage, Alaska, USA, 2011, pp.1394-1399

Urban Observatory (2013). http://www.urbanobservatory.org/ Accessed 1 September 2013

Wang, L., and Laszewski, G. (2008), 'Scientific Cloud Computing: Early Definition and Experience',hpcc, 2008 10th IEEE International Conference on High Performance Computing and Communications, pp.825-830

Weiss, A., (2007), 'Computing in the Clouds.' ACM Networker 11, no. 4, pp:18-25

You, P., Pen, Y., Liu, W. and Xue, S., (2012), 'Security Issues and Solutions in Cloud Computing,' 32nd International Conference on Distributed Computing Systems Workshops, Macau, 18-21 June 2012, pp. 573-577.

Chapter 4

4 Designing a Scalable Cloud Implementation for Mapping Geotagged Tweets

Abstract

The growth of social networks such as the Tweeter network has allowed access to message services through which hundreds of millions of people generate and share information about themselves, events, and places. The recent developments in the field of cloud computing have provided a way to process this massive amount of data in order to deliver on-demand services. In this chapter, a cloud based architecture is proposed in order to retrieve geotagged tweets, query its contents, and visualize the results on a map. The architecture consists of three main components, named as data collection, data processing, data visualization. For the data collection component, a Java application for pulling the Twitter's Streaming API was used to only retrieve geotagged Tweets. The Apache Hadoop ecosystem was used to store and query the retrieved geotagged tweets. The MapReduce, Hbase and Hive subprojects support the processing component of the proposed architecture. Finally, for the visualization component, the Mapbox's API and JavaScript were used to produce near-real time maps. The proposed architecture was evaluated and implemented using Twitter data collected from a recreational area in Vancouver, British Columbia.

4.1 Introduction

More than 550 million active users publish over 500 million 140-character "Tweets" every day (twitter statistics, 2013) and this makes tweets a new source of information that have found applications in disaster management (Earle et al., 2011), sentiment analysis (Parikh and Movassate, 2009), real time event detection and tracking traces (mobility). Moreover, with the increasing popularity of mobile applications such as Foursquare and the use of GPS systems in mobile phones, tweets have also become useful source of geo information since the tweets can be geotagged (also referred to geolocated or georeferenced). However, storing and processing geotagged tweets cannot be done using relational databases and traditional methods in GIS due to tweets' very large unstructured data volume as well as the need to achieve real time or near real time and scalable processing capabilities. It was this need that resulted in the development of cloud computing, which has been fundamentally driven by on-demand and pay-per-use services. As a result, cloud computing is expected to become the main framework in GIS in the near future in order to solve the processing challenges in handling geotagged Tweets.

In this Chapter, a cloud based architecture is proposed to collect process and visualize geotagged tweets. First, a brief literature review is provided to introduce cloud computing in GIS. Services of cloud computing as well as tools and technologies in implementing the case study are reviewed for all the three components of the architecture, advantages, challenges and obstacles of each component is discussed as well. In the next section, the proposed cloud architecture is described based on three

main components: data collection, data processing and data visualization. The implementation methods are discussed and the results from each component of the architecture are elaborated and presented and the final user interface of the implementation is presented. In the last section, the advantages and disadvantages of the implementation are discussed.

4.2 Cloud Computing in GIS

OGC's geospatial interface and standards had long been the *de facto* framework for presenting, processing and delivering geospatial information across Internet (OGC, 2013). But after the rise of big data and with the advancements in technology, a need for a new framework for processing and delivering geographical information started to emerge (Nie, 2011).

Cloud computing is a scalable framework consisting of a set of connected and virtualized computers over Internet, for enabling on-demand network access to a shared pool of configurable computing resources as a service (Mousavi and Wachowicz, 2013). Utilizing cloud computing, issues about handling big data as well as handling massive amount of geotagged Tweets are addressed and services are delivered over Internet with no limitation. Cloud computing has three main delivery models, Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS). The proposed architecture of this Chapter falls under the category of PaaS, but the author prefers to use the term "analytics as a service" due to the unclear definitions of the three

models, SaaS, PaaS and IaaS. A good example of PaaS is Google maps in which a platform is available and various types of information can be derived out of it.

4.2.1 Hadoop and MapReduce

Hadoop is an open source data management framework that has become widely deployed for massive parallel computation and distributed file systems in a cloud environment. Hadoop has allowed the largest web properties (Li, 2009) (Yahoo!, LinkedIn, Facebook, and Google) to store and analyze any type of data in near real-time and more efficient than traditional data management and data warehouse approaches could contemplate (Shvachco et al., 2010).

The Apache Hadoop software library is an open-source software framework that allows for the distributed processing of large data sets across large clusters of commodity hardware (White, 2009). Hadoop library enables applications to work with thousands of computation-independent computers and petabytes of data with capabilities of detecting and failing over at the application layer, so as to deliver a highly-available service on top of a cluster of computers, each of which may be prone to failures.

The entire Hadoop ecosystem consists of the Common kernel, HDFS (Hadoop Distributed File System), MapReduce (Dean and Ghewamat, 2004), as well as a number of related projects - among which Hbase and Hive are used for implementing the proposed architecture in this chapter. All of them are designed to facilitate processing in Hadoop and to make sure that node failures are automatically handled by the framework.

HDFS provides a distributed file system that stores unstructured data on the compute nodes with very high throughput across the cluster. Hadoop implements a computational paradigm for parallel processing of large data sets named MapReduce, where the application is divided into many small fragments of work, each of which may be executed or re-executed on any node in the cluster. HBase is a scalable, column-oriented, distributed database that supports structured data stored in large tables. By using HBase random, real-time read/write access to big data can be done. HBase leverages the distributed data storage on top of HDFS.

HDFS is used to store data especially for the semi-structured or unstructured data. Structured data like DBMS tables can be converted and stored into HBase. Also, spatial data such as vector data and image data can be stored in HBase. HBase provides an indexing mechanism by Row ID (rowed) for fast searching and acquiring of rows. Due to the large size and complexity of spatial data, traditional sequential computing models may take excessive time to do spatial analysis. MapReduce provides the capabilities of scaling down spatial data processing. For example, the parallel construction of image pyramid and other image processing by MapReduce from the boundless image data stored in HBase is widely used (Dean and Ghemawat, 2004).

Originally developed by Facebook, Hive (Thusoo et al., 2010) is an analytics tool, designed for ad hoc batch processing of potentially enormous amounts of data by leveraging map-reduce. Hive sits on top of a Hadoop cluster and provides an SQL like interface to the data stored in the Hadoop cluster. Hive can also effectively parallelize queries using MapReduce, a job that traditional single node databases cannot normally do.

The main reason of using Hive and Hbase together on top of Hadoop instead of just using Hbase is that Hive makes querying much easier than just using Hbase. Analyzing Hbase with MapReduce requires custom coding, and using Hive, simple SQL-like queries can be written to perform the same job.

4.2.2 Mapbox API

Data visualization has the ability to make the complex processing look simple and add extra value to the data. Intel states that if data visualization on big data is done to generate ideas and identify trends, it will add extra value to the big data and hence "Visualization" can be considered as the forth "V" alongside volume, variety and velocity as big data identifiers (UN Global Pulse, 2012).

According to IDG research services in 2012, top benefits of data visualization tools are: improved decision making, better ad-hoc data analysis and improved data sharing (IDG research services, 2013). The main reason Mapbox was chosen for the proposed architecture is that Mapbox has powerful tools for data visualization, it can easily handle big data results and the fact that Mapbox itself is a cloud based platform and is compatible with the proposed architecture.

It is also good to note that Leaflet (Leaflet library, 2013) is a contributor to Mapbox project, Leaflet is an open-source JavaScript library for mobile-friendly interactive maps, upon which Mapbox is basically built. As a result, the applications developed by Mapbox will all be mobile friendly which is an important feature of cloud based services.

4.3 Twitter API

Twitter is a massive social networking site tuned towards fast communication. More than 550 million active users publish over 500 million 140-character “Tweets” every day (Twitter statistics, 2013). Twitter is being used widely for social and political analysis and recently for geospatial analytic purposes (Villatoro et al., 2013). Geospatial analytics is being done because users of Twitter are tending more to share information about their geographical location using the GPS mounted on their smartphones or mobile applications such as Foursquare which automatically provide the geolocation of the user.

The two most used ways to access Tweets for developers is through Twitter’s streaming API and Twitter’s search API. In this section these two methods are briefly elaborated. Twitter’s Search API, involves polling Twitter’s data through a search or username. Twitter’s Search API gives access to a data set which already exists from Tweets. Through the Search API users request Tweets that match the “search” criteria. The criteria can be keywords, usernames, locations and named places. Twitter’s search API has its own limitations. With the Twitter Search API, developers can query Tweets which are limited by Twitter’s rate limits. For an individual user, the maximum number of Tweets which can be received is the last 3,200 Tweets, regardless of the query criteria. With a specific keyword, typically only the last 5,000 Tweets per keyword are accessible. Further limitations are by the number of requests which can be made in a certain time period. The Twitter request limits have changed over the years but are currently limited to 180 requests in a 15 minute period. And the most important limitation is the time limit of searching, based on the traffic of a server which the search

query goes to the oldest Tweets which can be retrieved are within the maximum of 14 days from the date of the query and this means that the historical data of Twitter can't be accessed through search API (Twitter search API, 2013).

Using Twitter's Streaming API, users register a set of criteria (keywords, usernames, locations, named places) and as Tweets match the criteria, they are pushed directly to the user in near real-time. The major drawback of the Streaming API is that it provides only a sample of Tweets that are occurring. The actual percentage of total Tweets users receive with Twitter's Streaming API varies based on the criteria users request and the current traffic. Studies have estimated that using Twitter's Streaming API users can expect to receive anywhere from 1% of the Tweets to over 20% of Tweets in near real-time (Twitter Statistics, 2013).

There are some libraries for parsing the Twitter API depending on the programming language which is used, for instance for Python, Tweepy (Tweetpy Project, 2013), and for Ruby, Grackle (Grackle project, 2013), for the development of the proposed architecture of this chapter Java language has been used along with Twitter4j (Titter4j project, 2013) library.

The Twitter Search API and Twitter Streaming API are widely used by researchers for light analytics or statistical analysis. For implementing this architecture both search and streaming API's have been used, however for industrial and commercial purposes, Firehose access (Twitter Firehose, 2013) should be granted.

4.4 The Architecture of the Implementation

In this section, the proposed architecture will be explained in detail. Figure 4.1 shows the overview of the architecture with the three main components of it: data collection component, data processing component and data visualization component.

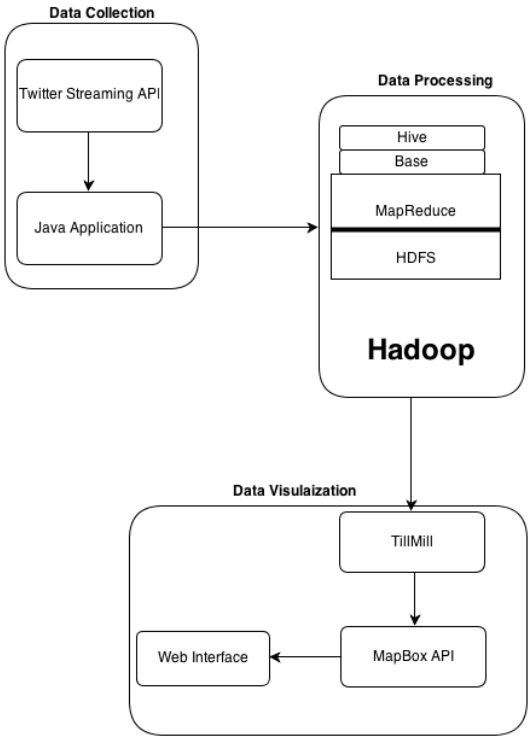

Figure 4.1: Overview of the General Architecture of the Implementation

4.4.1 Data Collection

Data Collection component is based on using Twitter's streaming API, to connect to Twitter's API and to be consistent with the processing part and Hadoop ecosystem. Java language and Twitter4j library have been used. Figure 4.2 shows in more detail the architecture of the data collection component. In general, the Java application has 3

responsibilities, establish the connection to Twitters API, get the Tweets with the defined filters and create the data set for processing.

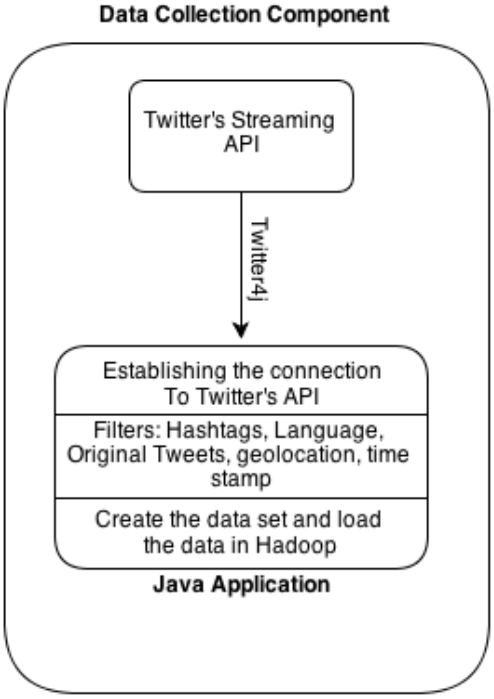

Figure 4.2: Architecture of the Data Collection Component

Accessing and storing Tweets are done in this component. After the connection is established to Twitter via Twitter's streaming API, a number of filters have to be defined to only collect the Tweets which meet the defined filters. Filters can be applied on information provided by the users such as name of profile, country of residence, postal code and etc. and all of this information can be retrieved along with the content of Tweets. After the Tweets are collected, then they can be stored in Hadoop's HDFS and be ready for processing and analyzing.

4.4.2 Data Processing

Data processing and data analysis is probably the most important component of the implementation since it refines the data and creates the desired result for visualization. As stated earlier, this component has been implemented using Hadoop ecosystem, Figure 4.4 shows the detailed architecture of this component.

Data which is being retrieved in near real time from Twitter's API are automatically imported to Hadoop ecosystem by the Java application and are stored in HDFS. Hbase has direct access to HDFS, and shares the HDFS with Hadoop, and as a result, technically it can be said that the data are being stored in Hbase. MapReduce is the processing framework and is the main reason for choosing Hadoop ecosystem for the development of this implementation. MapReduce is a programming paradigm that allows for massive scalability across hundreds of servers in a Hadoop cluster. But for this architecture, Hadoop has been mounted on a single machine (node) along with Hbase and Hive but the procedure which MapReduce executes stays the same. The term MapReduce actually refers to two separate tasks that Hadoop performs. The first is the map job, which takes a set of data and converts it into another set of data, where individual elements are broken down into key/value pairs. The input reader of the map job decomposes the data into small chunks and submits them to randomly chosen mapper programs. This process splits the input data and initiates the parallel processing stage.

After receiving the data, the mapper program executes a map function, and generates a collection of [key, value] pairs. Each produced item is sorted and submitted to the reducer. Then the reduce job starts and the reducer program collects all the items with

the same key values and invokes a reduce function to produce a single entity as a result. Figure 4.3 shows the MapReduce done on one machine (node), by adding extra nodes, the same paradigm continues, the input data will automatically be broken down and taken to new nodes in the servers and the new nodes will add extra power to processing to do the mapping and reducing jobs.

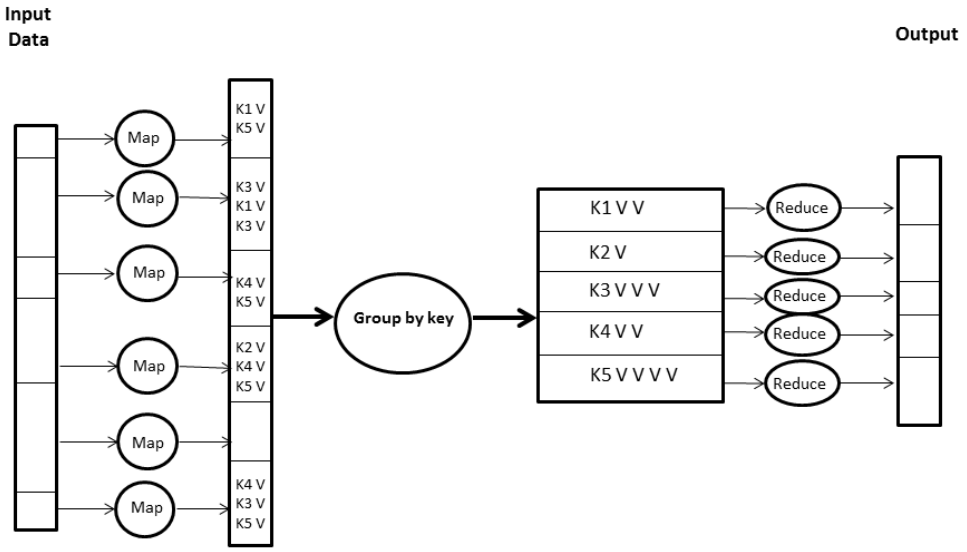

Figure 4.3 MapReduce Function on a Single Node

HBase is a column-oriented database management system that runs on top of HDFS. The reason Hbase is used in this ecosystem instead of Hadoop alone is mainly because Hbase does random reads and writes HDFS. If Hadoop is used alone, whenever MapReduce job runs the whole dataset will be read, but by using Hbase on top of Hadoop and by the random access that Hbase grants to the dataset, processing will be done much faster and in near real time (Figure 4.3).

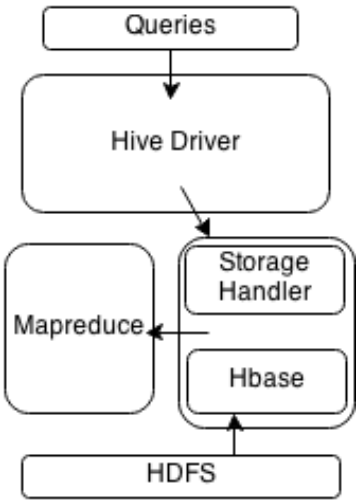

Figure 4.4: Hadoop Ecosystem with Hive and Hbase

Hive is basically a data warehouse. It sits on top of a Hadoop cluster and provides an SQL like interface to the data stored in the Hadoop cluster. SQL statements written in Hive interface are broken down by the Hive service into MapReduce jobs and executed across Hadoop cluster.

4.4.3 Data Visualization

The third and last component of the architecture is data visualization. After collected Tweets are processed in Hadoop, they are imported into TileMill (TileMill, 2013). TileMill is a design environment created by Mapbox (Mapbox, 2013), where the data is imported in TileMill in different layers. The number of layers in TileMill is the same as the number of layers visualized in the web application which in the case of the study area, it consists of 5 layers representing Tweets collected in the morning, afternoon, evening, night and the base map layer. As stated earlier, the Tweets have been collected

with their timestamps and as a result, the exact time in which a Tweet has been posted can be extracted. Therefore, results within the timestamp of 6am to 12 pm are considered morning Tweets and are imported to TileMill as well as the results for afternoon, evening and night Tweets. Then the styling has been done within TileMill using Cascading Style Sheets (CSS) standard. Results of each layer are uploaded to TileMill server as a web accessible tile. Using Mapbox API and leaflet library for JavaScript, the web interface is designed and displayed. Figure 4.5 shows the architecture of data visualization component in more detail.

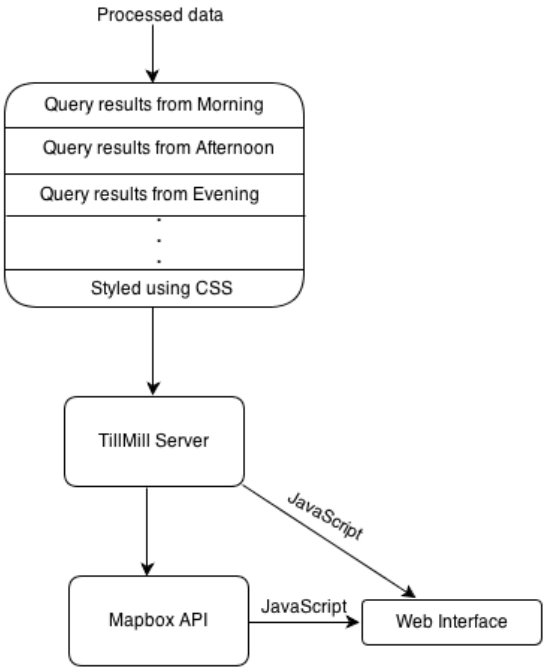

Figure 4.5: Data Visualization Component Architecture

4.5 Case Study

To evaluate the proposed architecture, a study area in Vancouver, British Columbia was chosen. The study area is called “Grouse Mountain” which is a recreation park located in northern Vancouver that is known for its steep trail (Grouse Grind).

4.5.1 Filters in Data Collection

Connecting to Twitter’s streaming API and collecting Tweets related to a location have been done using a number of queries. For the purpose of this research, only Tweets in English language were collected which have been tweeted as original Tweets (not re-Tweets or replies). The reason for applying these filters was the fact that the main application for this implementation is to provide the users with information on the activities taking place in a location. To be able to do this, in the processing component, words containing “ING” were searched for as an indicator for the activity in English language.

Evaluating the results containing re-tweets and replies, almost added no extra value to the data set as re-tweets were the copy of the tweets which have already been collected and replies were the same as re-tweets. As another filter, time stamp of the Tweets was also used, having the time stamp of the Tweets help categorize them more accurately in the visualization component.

One of the most important filters applied was the geolocation filter. The proposed architecture is for a geoweb application and as a result geotagged Tweets with known coordinates were retrieved.

It should also be noted that due to the privacy concerns user ID and names of the users have not been collected and the Tweets are anonymous. Instead of filtering the geographical location for the area of study, the hashtag (#) containing the name of the area of study was queried and filtered. It is also useful to note that according to Twitter Inc. in Twitter, hashtag symbol **(#)** is used before a relevant keyword or phrase in Tweets to categorize those Tweets and help them show more easily in Twitter search (Twitter Support, 2013).

The reason for querying hashtags rather than location is because of the purpose of the work. In the designed architecture we are interested to know the activities related to a location or a place, querying by location will not return all the related tweets about a specific location. As a matter of a fact, in the study case area in Vancouver, both methods were tested, and the results were as follows. When querying by location and coordinates, 4 coordinates (bounding box) in the vicinity of the area of interest was introduced to the Java application and around 50 percent of the returned Tweets were not talking about the specific location of the area of interest. But when querying by hashtags, more than 80 percent of the Tweets were exactly talking about the specific location. These results made clear that the decision for using hashtags was absolutely effective.

4.5.1.1 Collected Tweets for the Study Area

For the study area, 1011 Tweets have been collected by filtering #Grousegrind, #Grousemountain and #grouse and were imported to the Hadoop ecosystem. But to be

able to evaluate the functionality of the implementation more realistically, historical Tweets were needed, and as Twitter's API is unable to retrieve Tweets older than maximum 2 weeks, other sources were incorporated to provide historical Tweets. As a result TopSy (Topsy, 2013), a third party company who provides historical Tweets, was used and the Tweets starting from 2010 until October 2013 were collected for the test area using the same filters and criteria mentioned before.

Considering the small size of the area the number of retrieved Tweets from January 2010 to October 2013 were 1011, which is not a big number comparing to the actual number of Tweets being tweeted each day, but has its own set of advantages. One of the advantages of using a small dataset was the fact that the validity of using hashtags instead of bounding box as a filter could be tested and furthermore the effectiveness of querying "ING" for activity, as the key query of the methodology could also be evaluated.

4.5.2 Querying Tweets Using Hadoop

Hadoop ecosystem along with Hbase and Hive were used on one computer and in a virtual box on Windows operating system. The whole ecosystem has been made available and preconfigured by Cloudera (Cloudera, 2013) under "Cloudera Quick start VM". As the whole procedure was carried on a single computer, Hadoop ecosystem considered the job as a single node job (Figure 4.3).

As mentioned in the previous section, for activity monitoring "ING"s were queried in the dataset. 1011 Tweets have been captured and for the proof of validity of this

architecture, all the Tweets were reviewed carefully. In 1011 collected Tweets the total of 287 words ending in “ING” were detected among which 55 were referring words other than activities such as, Morning, King and everything. This shows that by querying “ING” 81% of the results refer to an activity done by the person who has tweeted it.

Using Hive interface on top of Hadoop with the data stored in HDFS and Hbase accessing the data, SQL queries were executed and the results were retrieved in near real time, as for the example below in 1.8 seconds (Figure 4.6, Figure 4.7).

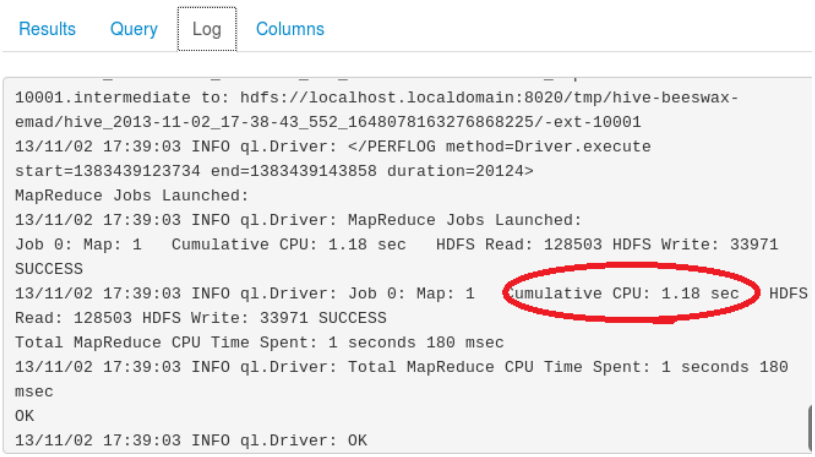

Figure 4.6: Time of Query Execution on the Whole Dataset Using Hive on top of Hbase

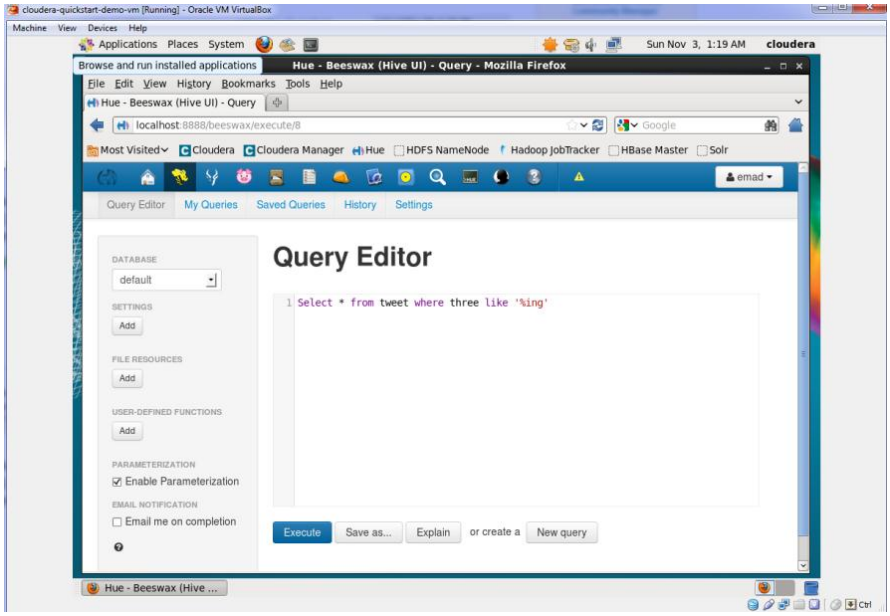

Figure 4.7: SQL Query in Hive for Retrieving Words Ending in ING

It is important to note again that Hive is used in the architecture as data querying tool and all the data is in Hbase and HDFS. Hive can use tables that already exist in HBase or manage its own ones, but they still all reside in the same HBase instance and Hive is used just as querying interface (Figure 4.3).

4.6 Data Visualization

Figures 4.8 and 4.9 show the web interface designed for this architecture, Activities are being shown in different times of a day, in the vicinity of Grouse Mountain and on the trail based on the geolocation of the Tweets.

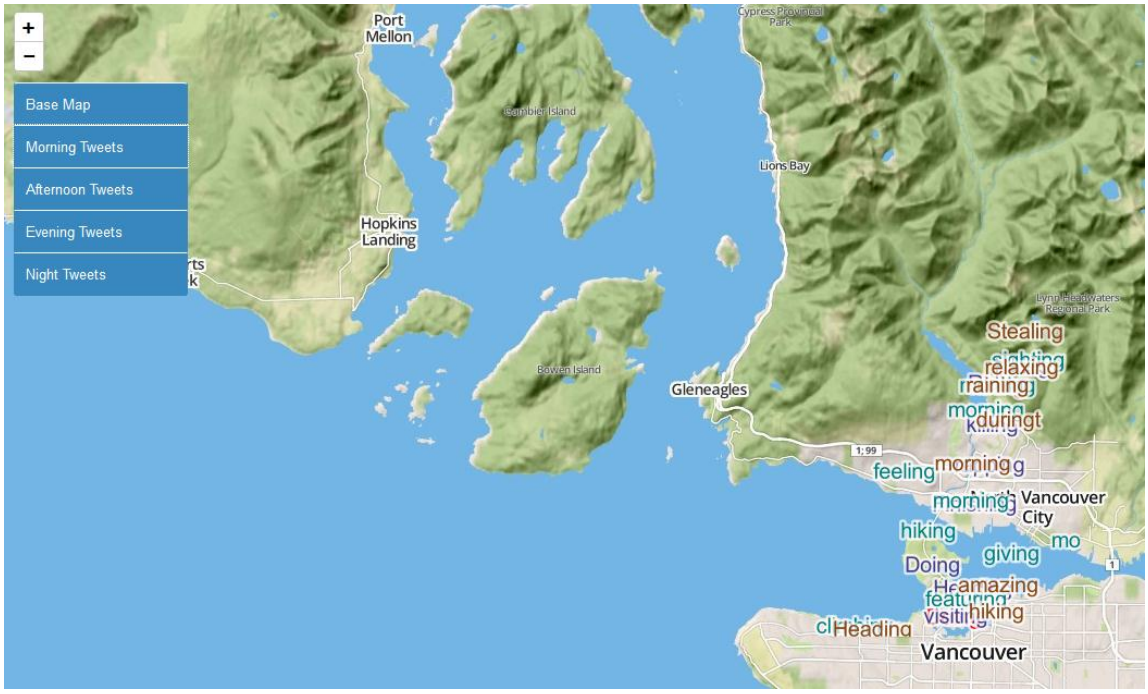

Figure 4.8: Overview of Displayed Tweets on Top of Base Map in Vancouver Area

The main contribution of this implementation is the application of the implementation which is mapping people's activities by connecting the virtual world of Twitter to the real world of maps. Users of social networks are increasing every day and this will help this implementation to portray an even better understanding of a place or location. Furthermore, mapping geolocated Tweets automatically gives an idea about the hotspots of an area. Hotspots are locations where people have more tendency to talk about them, and as a result, more Tweets are sent within those locations. Figure 4.9 and 4.10 shows the possible hot spots of Grouse Mountain.

Figure 4.9: Possible Hotspots of Grouse Mountain

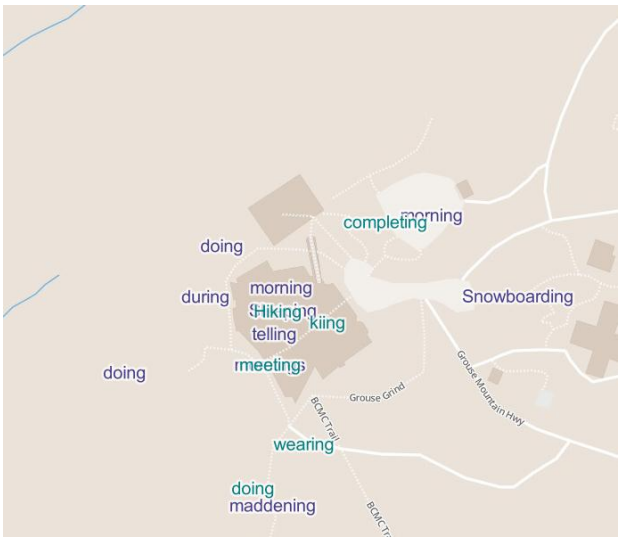

Figure 4.10: Possible hotspot of Grouse Mountain, Afternoon and Evening Tweets zoomed in

Tweets from different range of time are assigned a unique color so that the user can easily distinguish among the times. This enables users to know what activity is the most common in a certain time of day, and as result, makes it easier for the users to decide when to go to a place for a certain activity. Tweets from 6am to 12pm are morning Tweets and are colored red, tweets from 12pm to 6pm are afternoon Tweets and are

displayed in purple, Tweets from 6pm to 12am belong to evenings and are shown in blizzard blue and night. Tweets are brown and represent Tweets from 12am to 6am(Figure 4.11).

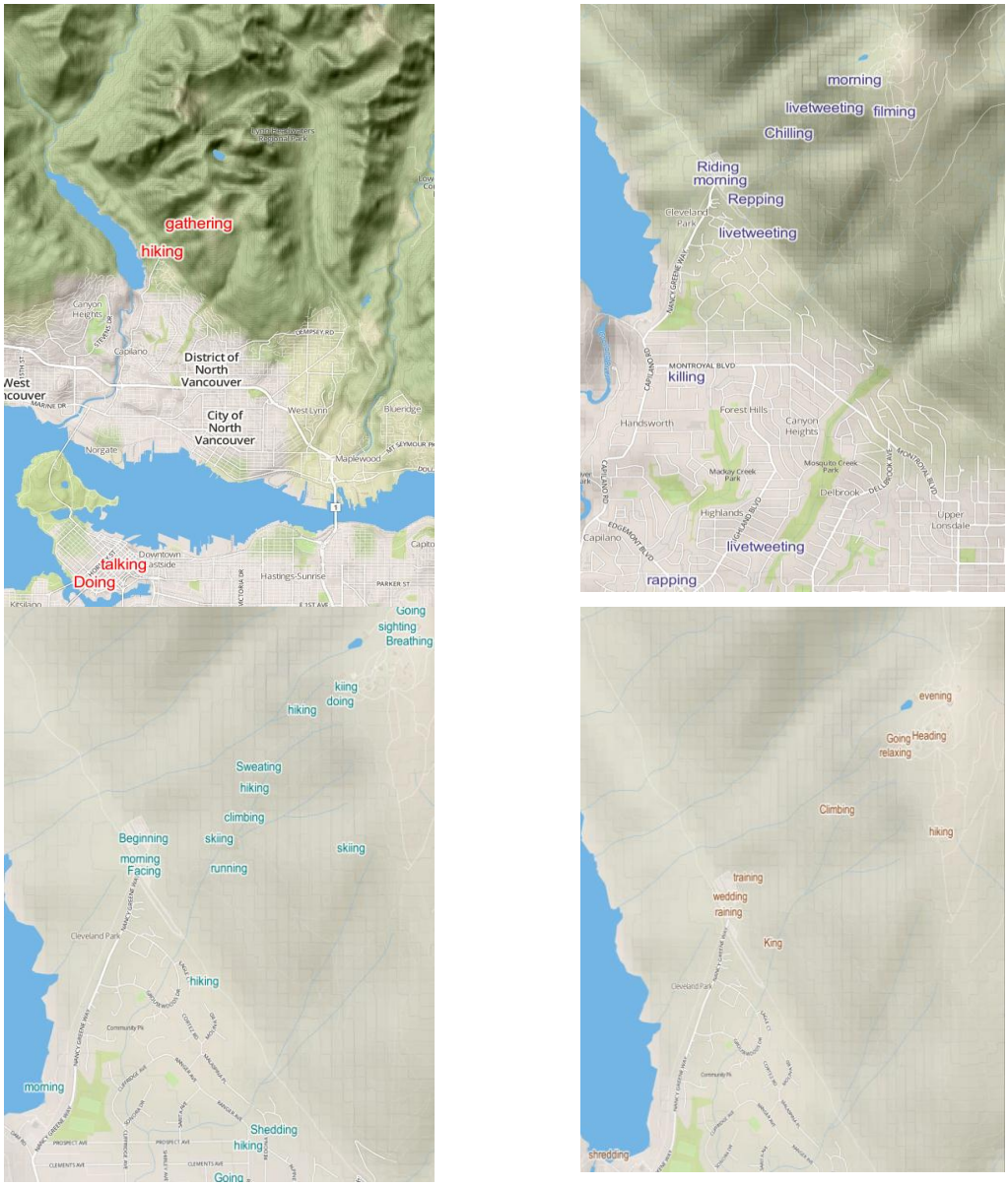

Figure 4.11: Morning, Afternoon, Evening, Night Tweets shown in Different Colors

Also by zooming in on the trail, the shape of the trail and the Tweets along the trail can be seen (Figure 4.12). The map in the web application is also dynamic, as a result by

zooming in, more tweets will be shown and the size of the displayed Tweets will change proportionally and automatically. The base map used in the web interface is the courtesy of Openstreetmaps.

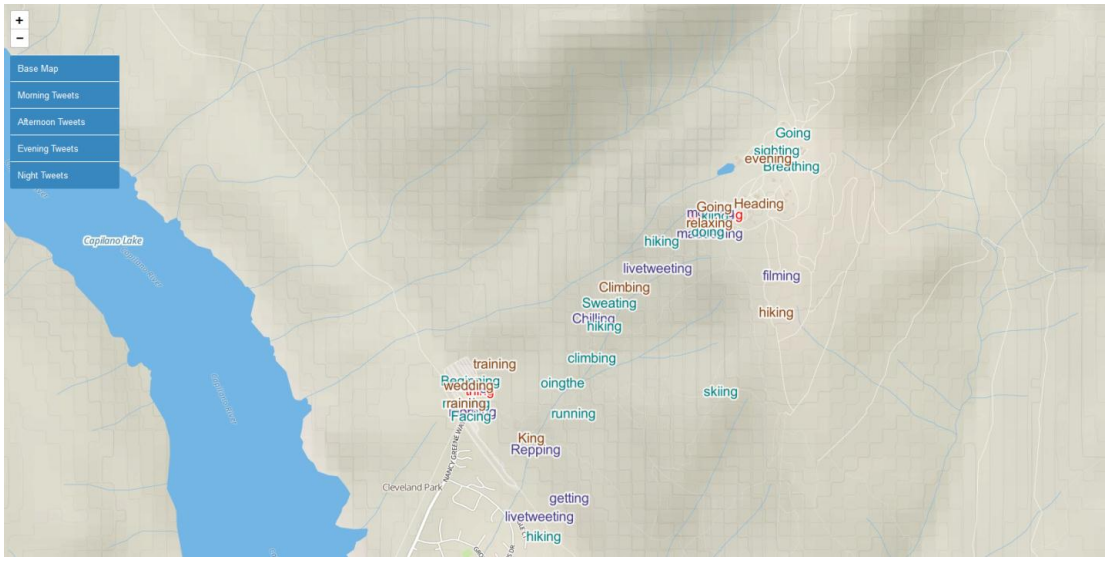

Figure 4.12: Tweets Along Grouse Grind Trail

4.7 Conclusions

In this chapter, a scalable cloud based architecture was designed and implemented. In this implementation activities of people who have Tweeted from the area of study have been extracted along with the time of which they have Tweeted, and have been visualized and mapped to provide more information and knowledge of that area and how people are connected to that place. Although this architecture has been tested for a small location but being a cloud based architecture, it is scalable and has the capability of

supporting huge amounts of data and could easily be extended for any place and location in the world having suitable map and data.

Hadoop ecosystem was used as the scalable cloud based processing engine and also Mapbox was used as a mobile and cloud based interface for visualizing the data which makes the architecture more scalable in terms of future expansions.

The main advantages of this implementation are: it is designed for handling geotagged Tweets, it is easily scalable for extra data and processing and the web interface is mobile and can be viewed on any device and computer and browser.

To further improve this architecture, the web interface can become more interactive by adding the option for users to choose among activities to visualize and to categorize the activities based on users' preferences. Adding these features make the architecture to be based more on the users' needs and thus one step forward in the goal for a comprehensive cloud based service.

References

Cloudera (2013), http://www.cloudera.com/content/cloudera/en/products-and-services/cloudera-enterprise.html/ Accessed 23 November 2013

Dean, J., and Ghemawat, S. (2004), 'MapReduce: Simplified Data Processing on Large Clusters', Sixth Symposium on Operating System Design and Implementation (OSDI04).

Earle, P., Bowden, D. and Guy, M. (2011), 'Twitter earthquake detection: Earthquake monitoring in a social world', Annals of Geophysics, volume 54, issue 6, pp. 708–715, and athttp://www.annalsofgeophysics.eu/index.php/annals/article/view/5364

Grackle project (2013). https://github.com/hayesdavis/grackle, Accessed 22 October 2013

IDG Research services white paper, (2013), 'Data Visualization: Making Big Data Approachable and Valuable', available at: http://www.sas.com/content/dam/SAS/en_us/doc/whitepaper2/sas-data-visualization-marketpulse-106176.pdf

Leaflet library (2013). http://leafletjs.com/ Accessed 25 October 2013

Li, P. (2009), 'Cloud Computing: Big Data Is The Future Of It', Accel Partners, available at http://assets.accel.com/5174affa160bd_cloud_computing_big_data.pdf

Mapbox (2013), https://www.mapbox.com/ Accessed 25 October 2013

Mousavi, S.E., Wachowicz, M. (2013), 'Cloud computing from perspective of GIS: An overview of impacts, benefits and technologies', Atlantic Canada conference 2013

Nie, N.H. (2011), 'The Rise of Big Data Spurs a Revolution in Big Analytics, Revolution Analytics Executive Briefing', available at http://www.revolutionanalytics.com/sites/default/files/the-rise-of-big-data-executive-brief.pdf

OGC (2013), http://www.opengeospatial.org/, Accessed 19 October 2013

Parikh, R. and Movassate, M. (2009), 'Sentiment Analysis of User-Generated Twitter Updates using Various Classi_cation Techniques' Stanford University

Shvachko, K., Huang, H., Radia, S., and Chansler, R. (2010), 'The Hadoop distributed file system, in: 26th IEEE Symposium on Massive Storage Systems and Technologies, May 2010.

Thusoo, A., Sarma, J.S., Jain, N., Shao, Z., Chakka, P., Zhang, N., Antony, S., Liu, H. and Murthy, R. (2010), 'Hive – A Petabyte Scale Data Warehouse Using Hadoop' In ICDE, pages 996–1005.

TileMill (2013). https://www.mapbox.com/tilemill/ Accessed 25 October 2013

Topsy analytics (2013) http://topsy.com/ Accessed 1 November 2013

Tweety project (2013)http://www.mthimm.de/projects/tweety/ Accessed 22 October 2013

Twitter Firehose (2013), http://apivoice.com/2012/07/12/the-twitter-firehose/ Accessed 24 September 2013

Twitter search API (2013), https://dev.twitter.com/docs/using-search/ Accessed 18 August 2013

Twitter statistics (2013). http://www.statisticbrain.com/twitter-statistics/, Accessed 11 October 2013

Twitter support (2013). https://support.twitter.com/articles/49309-using-hashtags-on-twitter# / Accessed 22 September 2013

Titter4j project (2013). http://twitter4j.org/en/index.html, Accessed 22 October 2013

United Nation global pulse, (2012), 'Big Data for Development: Challenges & Opportunities', available at: http://www.unglobalpulse.org/sites/default/files/BigDataforDevelopmentGlobalPulseMay2012.pdf

Villatoro, D., Serna, J., Rodríguez, V. and Torrent-Moreno, M. (2013). 'The TweetBeat of the City: Microblogging for Discovering Behavioural Patterns' during the MWC2012. Citizen in Sensor Networks (pp. 43-56)

White, T. (2009), ’A framework for large-scale data processing’, SoCC '12 Proceedings of the Third ACM Symposium on Cloud Computing, Article No. 4

Chapter 5

5 Conclusions and Future Work

This research was set out to find a scalable platform capable of storing, processing and visualizing large volumes of geospatial data and to portray the evolution of the architecture in computing paradigms. In the course of this study, two computing implementations, namely cloud computing and Internet computing, were studied, evaluated and tested.

For Internet computing, well-known WPS implementation were studied for their capabilities in raster processing. Geoserver proved to be the best implementation for raster processing. Also it was seen that using WPS even simple raster processing, can be very time consuming. Many factors affect the time of the process in WPS that should be taken into consideration. This makes WPS more complex to work with when dealing with large amounts of data. For example, Although the WPS implementation can be scaled up by adding extra machines and making a cluster of computers to improve the processing power, but the complexity in configuring the extra machines makes WPS a lesser suitable option for big spatial data processing platform.

For cloud computing, different definitions of cloud computing were studied and a new perspective towards cloud was proposed. A scalable architecture was designed and implemented to collect, process and map geotagged Tweets in order to display how people are connected to a location by mapping activities of people in that area. The platform was tested for a study area in North Vancouver, British Columbia.

This study identified that the cloud computing implementation is naturally the suitable architecture to design and implement scalable applications for the purpose of storing and processing geospatial data as the evolution of computing had indicated. This research also showed that GIS applications can be developed using cloud computing platforms.

5.1 Internet Computing

This study reviewed and examined 5 well-known implementations of WPS as Internet computing platforms for their capability in raster processing to identify whether WPS is a suitable option for developing a scalable platform in order to store and process big spatial data. The following were concluded:

- WPS platforms are implementations of OGC standard and as a result, developing applications based on these platforms are more comfortable and easier to be done than cloud computing platform which does not follow any standard.
- Although WPS platforms are capable of carrying out online processing, the run time of processing is usually high as it depends on the server and client's processing and storing power.
- WPS performs better when paired with desktop GIS such as Quantum GIS or ArcGIS.
- WPS platforms can be scaled up but the processing framework used in the platform is not designed for scalability and thus, it brings complexities in configuring and running the processing.

5.2 Cloud Computing

This research studied and examined cloud computing as the platform suitable for storing and processing big spatial data. Also the application and impacts of cloud computing and its effects on GIS and geospatial domain was tackled, and the following were discussed and concluded:

- Cloud computing is the result of evolution in computing paradigms and rise of big data is the main reason for the change in computing paradigms from Internet computing to cloud computing.
- Social networking platforms such as Twitter are examples of big data and to be able to handle, manage and process this big data, cloud computing frameworks and technologies need to be used.
- Existence of geotagged Tweets opens the path for researchers in geospatial domain to use this data as geographical data to develop applications based on the needs of users.
- Hadoop and its processing framework MapReduce are the de-facto of cloud based frameworks for storing and processing any type of data in any size.
- Although the designed architecture for the cloud computing platform have been implemented for a test area and on a single computer, using Hadoop and MapReduce, any number of computers can be added for scaling up without the need for configuration.

5.3 Research Limitations

The limitations of this research include the following:

- Due to time constraints and lack of resources, the implementation for cloud computing platform has been done on a single computer as opposed to a cluster of computers, also in WPS due to the same reason, only Geoserver had been implemented.
- As proof of concept, the data for cloud computing platform, geotagged Tweets, used in this study were a limited number to test the performance of proposed cloud based platform.
- As there was no previously tested architecture in literature for developing cloud computing architecture for processing geotagged Tweets, the efforts were mostly focused on studying and designing of the architecture, rather than implementing the platform. As a result, the platform is a prototype and can be improved from GUI point of view as well as processing.

5.4 Research Contribution

In the process of conducting this study, the following contributions have been made:

- In the course of this research a new synthesis has been made which was investigating and comparing Internet computing and cloud computing in regards

with implementation of Geoweb applications for storing, processing and visualizing large volumes of geospatial data.

- Two platforms have been studied and implementations have been made for both: Internet computing to collect and process raster data, and cloud computing to collect, process and visualize geotagged Tweets.
- A cloud application has been implemented for a study area in North Vancouver, British Columbia.
- A new approach in defining cloud computing has been made. That is defining cloud computing from the perspective of storing, processing and visualizing geospatial data.

5.5 Future Work and Recommendation

This study recommends the following for enhancing the results of this research:

- Continue the research on cloud computing by identifying the gaps of the proposed architecture and filling them up using new cloud based technologies.
- Improve and enhance the designed cloud based architecture, by developing a software or platform as a service.
- Increase input data for the platform as in collecting geotagged Tweets for more parks across Canada to incorporate in the platform for a testing the scalability of the architecture and methodology.

- Enhance the GUI of the cloud computing platform, making it more user-friendly by adding interactive features and properties.
- Further improvements on the platform can be made by making the platform a service, letting users choose which activities they want to display in a user chosen location.

Curriculum Vitae

Candidates Full Name

Seyed Emad Mousavi

Universities Attended

Zanjan University, Zanjan, Iran. B.Sc. in Geomatics Engineering

Publications and Conferences

Mousavi, SE., Wachowicz, M. (2013),'Applications of Cloud Computing in GIS', Geomatics Atlantic conference, Saint John New Brunswick, September 2013.

Mousavi, SE., Zhang, Y., Abouhamze, A. (2013),'Comparative Study of Web Processing Service Implementations for Raster Calculation', Canadian Institute of Geomatics Annual Conference (EOGC'2013), Toronto, Ontario, Canada, June 2013.

Mousavi, SE., (2012). 'Mosaicking and color balancing Landsat images using dynamic mosaicking In ArcGIS', Regional ESRI Users Conference, Fredericton, NewBruswick, November 2012.

Mousavi, SE., Abouhamzeh, A.,(2011). 'investigating and Analyzing Operating Systems of Smart Phones and Their Functioning in Providing GIS-based Services', FIG working week conference, Marrakech, morocco, May2011

Mousavi, SE., Abouhamzeh, A.,(2010). 'an automatic method for merging aerial and satellite images using fuzzy logic', 15th ARSPC conference, Alice Springs, Australia, 2010

Mousavi, SE., Abouhamzeh, A.,(2010). 'a study for choosing the best pixel surveying method by using pixel decisioning structures in satellite images', 24th international FIG conference, Sydney, Australia, 2010