Masters
Program
in **Geospatial**
Technologies

Parallelization of Web Processing Services on Cloud Computing: A case study of Geostatistical Methods

Carlos Andrés Osorio Murillo

Dissertation submitted in partial fulfilment of the requirements
for the Degree of *Master of Science in Geospatial Technologies*

UNIVERSIDADE NOVA ISEGI

UNIVERSITAT JAUME·I

WESTFÄLISCHE WILHELMS-UNIVERSITÄT MÜNSTER

Parallelization of Web Processing Services on Cloud Computing:

A case study of Geostatistical Methods

By

Carlos Andrés Osorio Murillo

Supervised by

Joaquín Huerta Guijarro, Ph.D

Departamento de Lenguajes y Sistemas Informaticos, Universitat Jaume I, Castellón, Spain.

Cosupervised by

Albert Remke, Ph.D

Institute for Geoinformatics, Westfälische Wilhelms-Universität, Münster,Germany.

and

. **Marco Painho, Ph.D**

Instituto Superior de Estatística e Gestão de Informação, Universidade Nova de Lisboa, Lisbon, Portugal.

February 2011

AUTHOR'S DECLARATION

I hereby declare that this thesis has been written independently by me, solely based on the specified literature and resources which have been cited appropriately. The thesis has never been submitted for any other examination purposes. It is submitted exclusively to Universities participating in the Erasmus Mundus Master program in Geospatial Technologies.

Castellon de la Plana, 28 February 2011

Carlos Andres Osorio Murillo

ACKNOWLEDGMENTS

I would like to thank my supervisor Dr. Joaquin Huerta, for the interest and appreciation in my work, and co-supervisors Dr. Albert Remke, and Dr. Marco Painho for their support and guidance throughout the process of writing this thesis. Besides, I appreciate the help received by Dr. Carlos Granell and Dr. Laura Diaz, of the Universitat Jaume I, including their advice, and ideas to improve my work. And a special thanks to Dolores Apanewicz, the Master Administrator, for her assistance during this time, and for her revision of my thesis, to improve its writing.

I would like to thank the European Commission and the Erasmus Mundus Master in Geospatial Technologies staff for having been selected and granted the scholarship.

Finally, I would like to thank my family, and, I wish to express my special gratitude to my wife for her loving support, continuous encouragement and for never stopping believing in me.

Parallelization of Web Processing Services on Cloud Computing: A case study of Geostatistical Methods

ABSTRACT

In the last decade the publication of geographic information has increased in Internet, especially with the emergence of new technologies to share information. This information requires the use of technologies of geoprocessing online that use new platforms such as Cloud Computing. This thesis work evaluates the parallelization of geoprocesses on the Cloud platform Amazon Web Service (AWS), through OGC Web Processing Services (WPS) using the 52North WPS framework. This evaluation is performed using a new implementation of a Geostatistical library in Java with parallelization capabilities. The geoprocessing is tested by incrementing the number of micro instances on the Cloud through GridGain technology.

The Geostatistical library obtains similar interpolated values compared with the software ArcGIS. In the Inverse Distance Weight (IDW) and Radial Basis Functions (RBF) methods were not found differences. In the Ordinary and Universal Kriging methods differences have been found of 0.01% regarding the Root Mean Square (RMS) error.

The parallelization process demonstrates that the duration of the interpolation decreases when the number of nodes increases. The duration behavior depends on the size of input dataset and the number of pixels to be interpolated. The maximum reduction in time was found with the largest configuration used in the research (1.000.000 of pixels and a dataset of 10.000 points). The execution time decreased in 83% working with 10 nodes in the Ordinary Kriging and IDW methods. However, the differences in duration working with 5 nodes and 10 nodes were not statistically significant. The reductions with 5 nodes were 72% and 71% in the Ordinary Kriging and IDW methods respectively.

Finally, the experiments show that the geoprocessing on Cloud Computing is feasible using the WPS interface. The performance of the geostatistical methods deployed through the WPS services can improve by the parallelization technique. This thesis proves that the parallelization on the Cloud is viable using a Grid configuration. The evaluation also showed that parallelization of geoprocesses on the Cloud for academic purposes is inexpensive using Amazon AWS platform.

Paralelización de Web Processing Services en Cloud Computing: Un caso de estudio en métodos geostadísticos

RESUMEN

En la última década la publicación de la información geográfica se ha incrementado en Internet, especialmente con la aparición de nuevas tecnologías para compartir información. Esta información requiere el uso de tecnologías de geoprocesamiento en línea que utilizan nuevas plataformas como Cloud Computing. Esta tesis evalúa la paralelización de geoprocesos en la plataforma Cloud de Amazon Web Service (AWS), mediante OGC Web Processing Services (WPS) usando la aplicación de 52North. Esta evaluación se realiza mediante la implementación de una nueva biblioteca geoestadística en Java con capacidades de paralelización. El geoprocesamiento es probado incrementando el número de nodos (micro instancias) en la plataforma Cloud a través de la tecnología GridGain.

La biblioteca geoestadística obtiene similares valores interpolados en comparación con el software ArcGIS. En los métodos de Ponderación del Inverso de la Distancia (IDW) y Función de Base Radial (RBF) no se encontraron diferencias. En los métodos Kriging Ordinario y Kriging Universal se encontraron diferencia de 0.01% con respecto al error medio cuadrático.

El proceso de paralelización demuestra que la duración de la interpolación disminuye cuando el número de nodos aumenta. El comportamiento de la duración del proceso depende de la cantidad de datos de entrada y el número de pixeles a interpolar. La reducción máxima de tiempo se encontró con el conjunto de datos mas grande utilizado en la investigación (1.000.0000 de pixeles y un conjunto 10.000 puntos). El tiempo de ejecución disminuyo en 83% trabajando con 10 nodos en los métodos Kriging Ordinario e IDW. Sin embargo, las diferencias en la duración trabajando con 5 nodos y 10 nodos no fueren estadísticamente significativas. Las reducciones con 5 nodos fueron 72% y 71% en el Kriging Ordinario e IDW respectivamente.

Finalmente, los experimentos muestran que el geoprocesamiento en Cloud Computing es factible a través de la interface WPS. El rendimiento de los métodos geostadísticos desplegados mediante los servicios WPS puede mejorar con la técnica de paralelización en el Cloud. Esta tesis prueba que la paralelización de geoprocesos en Cloud Computing para propósitos académicos no es costosa usando la plataforma Amazon AWS. Todavía

KEYWORDS

Web Processing Services

Parallelization Algorithms

Interpolation

Geostatistics

Cloud Computing

PALABRAS CLAVES

Web Processing Services

Algoritmos de paralelización

Interpolación

Geoestadística

Cloud Computing

ACRONYMS

Amazon RDS	Amazon Relational Database Service
Amazon S3	Amazon Simple Storage Service
AMI	Amazon Machine Image
API	Application Programming Interface
AWS	Amazon Web Services
BPEL	Business Process Execution Language
DEM	Digital Elevation Model
EBS	Elastic Block Store
EC2	Amazon Elastic Compute Cloud
GAE	Google App Engine
GI	Geographic Information
GIS	Geographic Information Systems
GML	Geographic Markup Language
GPW	The Geo Processing Workflow
GRAM	Globus Resource Allocation Manager
GSI	Grid Security Infrastructure
HaaS	Hardware as a Service
HDFS	Hadoop Distributed File System
IaaS	Infrastructure as a Service
IDW	Inverse Distance Weight
KML	Keyhole Markup Language
KVP	Key Value Pairs
OGC	Open Geospatial Consortium
PaaS	Platform as a Service
RMS	Root Mean Square

SaaS	Software as a Service
SDI	Spatial Data Infrastructures
SOA	Service-oriented architecture
SOAP	SOAP
SPI	Service Provider Interface
UNICORE	Uniform Interface to computing Resources
UTM	Universal Traversal Mercator
WCS	Web Coverage Service
WFS	Web Feature Service
WMS	Web Map Service
WPS	Web Processing Service
WSDL	Web Services Description Language
XML	eXtensible Markup Language

INDEX OF THE TEXT

INDEX OF TABLES

INDEX OF FIGURES

1. Introduction

In the last years Internet has changed the face of applications and the environment in which they are executed. Everyday there are more online applications that offer the same tools that were offered by desktop applications. The mechanisms used to manipulate, share and generate geographic information (GI) are also changing. Nowadays, Spatial Data Infrastructure (SDI) technology is contributing to the implementation of new methodologies for improving the manipulation of GI at different levels in our society. One of the most important points in building SDI is the adoption of standards for sharing GI. Thus, the Open Geospatial Consortium (OGC) is becoming an important part of SDI with standards such as Web Map Services (WMS), Web Feature Services (WFS), Web Coverage Services (WCS), Web Processing Services (WPS) and others.

The WPS standard increases the potential of geoprocessing online through publication of tools already developed in Geographic Information Systems (GIS) software or procedures that incorporate complex processes (Ladra et al., 2008). The performance improvement of WPS services is an important theme in the development of the OGC interface (Brauner et al., 2009). Technologies such as Grid computing are being evaluated to improve the specification (Baranski, 2008). This technique uses the parallelization of processes to execute a complex task. Using the features of Grid computing in geospatial data it is possible to improve the performance of WPS services on Internet. This thesis combines the parallelization technique that is generally used in Grid computing to interpolate GI through OGC services.

The Grid computation paradigm uses a network of dedicated servers to solve a particular problem i.e., Search for Extraterrestrial Intelligence (SETI). This approach is not applicable to GI. It is not possible to dedicate a network of servers to solve a simple geographical problem. The concept of Grid computing to solve multiple problems, simultaneously and focus on users has to be adapted. This concept is incorporated on the new paradigm called Cloud Computing.

Cloud Computing combines some features such as virtualization, high potential, low cost and service oriented (Zhang et al., 2010a) that incentives the development of a new model for processing, storing and sharing information. The GI is also included in the type of information suitable for the Cloud Computing environment. The best known geographic application on Cloud Computing is Google maps (Velte et al., 2010) which is used by thousands of people every day. The OGC standard Keyhole Markup Language (KML) is being used to share geographic information, and its expansion requires the implementation of applications that support it; but, the amount of data generated is a problem for geoprocessing. The development of technologies that process information in Internet is needed to avoid the data problems. There are several types of generators of GI such as GPS, sensors, weather stations, and others that require geoprocessing on line. Usually, GI is related with continuous variables that require specialized software applications or techniques like Geostatistics.

Geostatistics is used to determine the best spatial distribution of a variable. This technique uses interpolation for predicting and evaluating the behavior of a variable. It is used in different areas like agriculture, climatology, business, topography and others. This project implements some Geostatistical methods through a Java library. This library has been designed to be executed in parallel with WPS services, which are deployed on Cloud Computing with some capabilities of Grid computing. This project evaluates the performance of execution of interpolations on Cloud Computing.

1.1 Problem statement

The geoprocessing of GI on Internet requires the transmission of large datasets to be processed. The paradigm of downloading the data to be processed is currently changing. Every day, there are more Cloud applications for storing, processing and analyzing information without having to download it. This thesis work contributes to the evaluation of Cloud Computing for geoprocessing on Internet, using the interface WPS with parallelization capabilities

1.2 Objectives

The major goal of this work is the evaluation of the parallelization of geoprocessing on the Cloud Computing through the WPS interface.

- Implement a Geostatistical library with parallelization features in order to reduce the duration of calculations.
- Generate WPS services with the parallel capabilities to process information on Grid.
- To evaluate the feasibility of geospatial analysis on the Cloud through parallelization of geoprocessing

1.3 Thesis structure

The first chapter introduces the general information and the objectives of this research. The second chapter provides the theoretical background about the main topic of the thesis: WPS, Geostatistics and Cloud Computing. The third chapter describes the dataset, software and hardware used in the project. The Geostatistical methods library is presented in the fourth chapter, which indicates all aspects involved in the design and implementation of the Geostatistical library in Java. The implementation of the Geostatistical library in the WPS is described in the fifth chapter. The sixth chapter shows the steps followed in the implementation on the WPS on the Cloud. Chapter seven describes and discusses the results obtained in the project. Finally, chapter eight provides the conclusions and future work.

2. **Background**

2.1 Web Processing Service (WPS)

The expansion of GIS technology and geographic data through different areas has created the need of sharing, and exchanging geographic information among producers of geographic information and users. The Open Geospatial Consortium (OGC) works in the generation of open standards that facilitate the communication and processing of geographic information (OGC Reference Model, 2008). The areas in which OGC works are related with the access and process of geodata, creation of interfaces, and consensus of methodologies for interoperability. The OGC web services are based on an open non-proprietary Internet Standard specification (OGC Reference Mode, 2008), in order to support geodata, geo-processes, sensors, location services and other services related to geographic information. The OGC WPS was accepted as a standard interface that allows wrapping a process, algorithm or operation on Web in a defined structure, which can be discovered and used by others processes or clients (OGC Web Processing Service, 2007). The WPS describes the inputs and outputs of the processes and mechanisms that should be used by a request to obtain a result. This allows integrating and binding any type of format and procedure. Each WPS service has an identifier in order to facilitate its discovery.

The WPS has been used in projects related with disaster management in urban areas (Stollberg & Zipf, 2009), that allows combining several data sources and process chaining to determine risk areas. In others fields like precision agriculture, it has been used to support decision making of farmers (Nash et al., 2007). Some hydrological projects have incorporated the WPS specification to model watersheds (Fitch & Bai, 2009; Díaz et al., 2008). These projects have demonstrated the usefulness of the specification on complex geoprocessing workflows. However, they suggest working in problems related to the support of different Geographic Markup Language (GML) versions and huge datasets management.

2.1.1.1 WPS operations

The WPS establishes three mandatory operations that can be managed by a XML-based protocol with a POST method and Key Value Pairs (KVP) with GET method. The WPS specification version 1.0.0 supports the Simple Object Access Protocol (SOAP) to exchange structured information. This new feature allows integrating WPS with Service-oriented architecture (SOA) in order to improve the interoperability with others systems.

- GetCapabilities: This operation retrieves the relevant information about the service provider, and describes all the processes available by the service.

- GetDescription: This operation is usually executed after the GetCapabilities operation to describe a particular process. This operation uses the process identifier to obtain information about inputs and outputs identifiers, and all needed schemas to be recognized by a server and a user understandable description. This operation also provides the supported formats and optional values that each input can have.

- Execute: This operation requires the process identifier and the value of each parameter in the supported format. The output of the operation is a XML-Document with a description of the process and the outputs. The outputs can be literal data e.g., String, Double, Integer and etc., and complex data as GML document, compressed Shapefile, GeoTiff and so on.

2.1.1.2 WPS Implementations

The OGC WPS standards have had several versions from 0.4.0 to the current version 1.0.0 which was released in 2007 (OGC Web Processing Service, 2007). During the development of this standard some projects have worked on supporting new versions, including modifications and improvements to obtain a final complete version. Some projects that support the WPS version 1.0.0 specification are:

- Deegree[1]: This project supports the complete implementation of WPS 1.0.0 specification and KVP, XML and SOAP requests. The application is deployed through a ServletContainer on TOMCAT or Jetty.

- PyWPS: This project is based on Python and provides native support for GRASS GIS using the WPS 1.0.0 specification. This server is designed to deploy processes of other software, like R statistic, GDAL or PROJ (PyWPS, n.d.).

- 52 North WPS[2]: The implementation of 52North supports the WPS specification version 1.0.0 through the use of Java technology. This framework uses Geotools libraries to manage geographical geometries and complex data. It also includes some extensions to support several types of processes providers e.g., GRASS, Sextante, and connection with ArcGIS Server. This work tests new features such as extensions

[1] www.deegree.org
[2] http://52north.org/maven/project-sites/wps/52n-wps-site/

in the implementation, for example, transactional profile and process parallelization using UNICORE or GridGain.

2.1.1.3 WPS on Grid computing

The concept of Grid computing is related with two problems, the addition of processing power and the distribution of resources (Zhang et al., 2010b). Usually, Grid computing implies the division of a procedure for getting better performance in the execution of a process. Brauner et al. (2009) has argued that efficiency of geoprocessing services is an important topic in which the community should work to improve the WPS standard. In this way, technologies as parallel processing, distributed algorithms and agent-based modeling (Yuan, 2007) can improve the performance of geoprocesses. Although at this moment, the specification does not fully support geoprocesses on Grid as shown by Baranski (2008), the Grid profile is being studied by OGC.

In a Grid environment the geoprocesses should use the technique of parallelization of algorithms, which can be classified in two types: simple parallelization and data parallelization (Pautasso & Alonso, 2006). The simple parallelization technique divides the problem by using threads of controls, in which there is a dependency during the execution. Otherwise, the data parallelism is often used over large datasets. This method splits the dataset into subsets and executes an operation independently for each one (Pautasso & Alonso, 2006). The data parallelism is divided into:

- Static: nodes' number is known before execution
- Dynamic: nodes' number is obtained at runtime
- Adaptive: the tasks' number is calculated based on number of nodes. The adaptive approach also depends on data homogeneity and its relation with task duration in each node (Mahanti & Eager, 2004).

The execution of parallel processes requires a framework which manages problems associated with the distribution of tasks. There are some open grid frameworks that provide support for Grid infrastructures as GridGain, Hadoop, Globus Toolkit, Unicore and etc.

- GridGain: This framework is based on Java technology, and it improves the performance of an application dividing and parallelizing tasks. It also allows managing the Grid topology through the Service Provider Interface (SPI). This SPI helps to distribute all processes adequately on the nodes, and manages failures on transactions among nodes (Resende, 2010*)*. This technology has been evaluated on the implementation of WPS 52North, in which the essential libraries of GridGain has been added. The last version 3.0 supports the auto scaling of a Cloud and other characteristics such as: Cache distributed data in data grid, auto-discover all grid resources and scale up or down based on demand (GridGain, 2010).

- Globus Toolkit: This is a set of open tools to build grids. It has some principal modules: Globus Resource Allocation Manager (GRAM) which allows for controlling, executing and supervising jobs and Grid Security Infrastructure (GSI) to improve the security on all levels of the grid. Also, it includes tools for resources management, fault detection, communication, and portability. This project is adopted by several institutions such as the University of Chicago, NASA, DARPA, IBM and Microsoft[3].

- Uniform Interface to computing Resources (UNICORE): "Make distributed computing and data resources available in a seamless and secure way in intranets and the Internet[4]", this project has been used in the WPS framework of 52North to demonstrate the capabilities of parallelization of processes using the interface.

- Hadoop: This framework allows for the management of a large amount of data in parallel, this technology uses the principles of a MapReduce technique. This programming technique divides the process in two sections, Map and Reduce. In the Map, a central node splits and distributes the input into small parts, each part is worked independently. The Reduce section is in charge of obtaining the responses of all the nodes. The input and worked part is stored in the Hadoop Distributed File

[3] http://www.globus.org/toolkit/about.html

[4] http://www.unicore.eu/index.php

System (HDFS)[5]. The MapReduce technique is implemented on multiple projects of Google (Dean & Ghemawat, 2004). According with Ku et al. (2010), it is possible to use this technology with massive geodata, and through WPS, the operations built up in this system can be accessed.

The WPS on Grid computing has been evaluated by Pascoe et al. (2009) in the calculation of global and regional climate models, designed to support 1000 simultaneous request over WPS layers. Other projects have used parallelization techniques to improve the management of images and interpolations (Alonso-Calvo et al., 2010; Hawick et al., 2003; Pesquer-Mayos, 2008).

2.1.1.4 Orchestration of WPS

According to Brauner et al. (2009) the orchestration or workflow of WPS is an essential topic to improve the WPS specification. The OGC is also investigating a new specification to manage workflows. The Geo Processing Workflow (GPW) is a new approach that works with Business Process Execution Language (BPEL) and Web Services Description Language (WSDL) to orchestrate OGC WPS (OWS-4 Geo Processing Workflow (GPW)). It is not possible to include directly WPS on orchestration model with BPEL. Its description should be converted to a WSDL document (Stollberg & Zipf, 2008). The combination of WPS and WSDL improves the reusability. When the WPS binds complex processes it can lack reusability and flexibility (Wehrmann et al., 2010).

2.1.2 Others OGC Web Services

The OGC classified Web Services depending on their functionalities to manage geospatial data, process information, sensor management, and mass services. The OGC Web Services projects that focused on geodata are:

- Web Map Service (WMS): This service provides some mechanisms to share geodata visually using three operations; GetCapabilities, GetMap and GetFeatureInfo. With these operations it is possible to obtain and overlay data of

[5] http://hadoop.apache.org/mapreduce/docs/current/mapred_tutorial.html#Purpose

diverse platforms and sources. The GetCapabilities is a common denominator in the OGC Web Services that manage geodata.

- Web Feature Service (WFS): The OGC represents the geodata using Geographic Markup Language (GML), which allows modeling any geographic element. The WFS is a service that provide mechanisms to manage geographical features using GML formats through transactional operations such as insert, update and delete.
- Web Coverage Service (WCS): The grid structure represents information usually provided by satellite images, Digital Elevation Models (DEM), and other kinds of geographical information sources. The OGC have developed the WCS standard to facilitate the manipulation of raster information in a Web environment.

2.2 Geostatistics

The Geostatistics term describes some statistical methods applied in a geographic context. Usually they use continuous variables that can be measured anywhere. These methods are also associated with some interpolation techniques as Kriging, Inverse Distance Weight, Spline, etc. These methods share a similar objective, to obtain an unknown location value from known values of other locations. The methods suppose that the unknown value is a combination of weights and known values. The general equation (1) describes the combination of weights λ, and known values z to obtain the unknown $z_{0.}$ The distance plays an important role in the determination of each weight and each technique has its own form to obtain the weights.

$$z_o = \sum_{i=1}^{s} \lambda_i \, z_i \tag{1}$$

The interpolation methods can be classified by assessment of error in deterministic or stochastic methods; by points used in global or local; or by exactitude in exact or inexact. In the table 1 the characteristics of the methods used in this project are described.

Local interpolation methods	
Deterministic	Stochastic
Inverse Distance Weight (Exact)	Ordinary Kriging (Exact)
Spline (Radial basis functions) (Exact)	Universal Kriging (Exact)

Table 1 Characteristics of local interpolation methods

2.2.1 Geostatistical Methods

2.2.1.1 Inverse Distance Weight (IDW)

This is a local, deterministic, and exact spatial interpolation method which is frequently applied on geosciences (Chang, 2004). IDW method suggests that the attribute values of two points are related by the inverse of their distance. Lu & Wong (2008) states that it is usual to modify the distance weight to predict the value of an unknown attribute of a location. The unknown value is calculated by the equation (1).

$$z_o = \frac{\sum_{i=1}^{s} z_i \frac{1}{d_i^k}}{\sum_{i=1}^{s} \frac{1}{d_i^k}} \tag{2}$$

Where, z_0 is the value to be estimated at point 0, z_i is the value at a known point i, d_i is the distance between a known point i and point 0, s is the number of points used and k is the power used. The equation can be represented by the general equation (1) where, each weight λ is calculated by the equation (3).

$$\lambda_i = \frac{\frac{1}{d_i^k}}{\sum_{i=1}^{s} \frac{1}{d_i^k}} \tag{3}$$

2.2.1.2 Kriging

This method is related with the definition of spatial correlation. Its principal assumption is a stationary approach, in which the relationship between values of whatever pairs of points is independent of their position and the covariance is similar in all the points that are at the same distance (Johnston et al., 2001). This relation is managed through an empirical semivariance. This method searches for the best theoretical semivariogram model (Appendix A) to fit the empirical semivariogram data. In this way, it is possible to obtain the error estimation. Kriging has some variations that depend on the type of data, some presumptions as normality, knowledge of the mean or the tendency of data. This research works with two types: Ordinary and Universal Kriging.

The Kriging methods use the empirical semivariance to model the behavior of data. Some predefined theoretical models are used to fit the characteristics of empirical semivariance. These mathematical models can be defined by three parameters: range, sill and nugget (figure 1). The variability of the semivariogram is defined by the range. After this value the semivariance is constant. The sill defines the semivariance threshold. When the variability in the semivariogram is not explained by just the sill, the evaluation of a nugget effect it is needed. Finally, the empirical semivariogram is evaluated on intervals called lags. The number of lags and the length of lags can influence the behavior of the theoretical semivariogram.

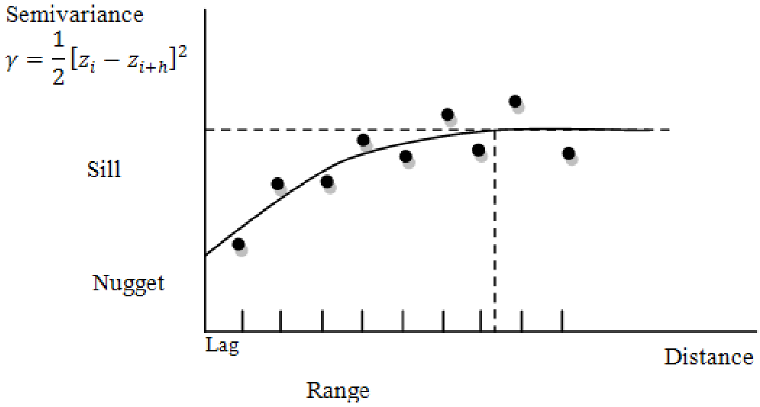


Figure 1. Empirical and theoretical semivariogram

Ordinary Kriging assumes that autocorrelation among all points range *h* is the average semivariance (Chang, 2004) show by equation (4)

$$\gamma(h) = \frac{1}{2n}\sum_{i=1}^{n}[z(x_i) - z(x_i + h)]^2 \tag{4}$$

The unbiased condition in Kriging defines that the expectation of errors should be zero; it is showed in the equation (5).

$$E(z_o - \sum_{i=1}^{s} \lambda_i\, z_i) = 0\,; \tag{5}$$

The λ values are obtained minimizing the least square of the equation (6).

$$E(z_o - \sum_{i=1}^{s} \lambda_i\, z_i)^2 = minimizing\,; \tag{6}$$

$$\begin{bmatrix} \Sigma_z & 1 \\ 1' & 0 \end{bmatrix} \begin{bmatrix} \lambda \\ m \end{bmatrix} = \begin{bmatrix} c \\ 1 \end{bmatrix} \tag{7}$$

In the equation (6) (Johnston et al., 2001) the sum of all λ values should be equal to 1. In this case, it is needed to use the Lagrange multiplier *m*. Σ_z represents the theoretical semivariance matrix (equation 7), and *c* the values of the unknown semivariance calculated through theoretical model. The λ values are replaced in the equation (1) to get the unknown value z_0.

Universal Kriging assumes that autocorrelation among all points range *h* is affected by a tendency (Chang, 2004) shown by equation (8), where β represent the trend.

$$z_o = \sum_{i=1}^{s} \lambda_i\, \beta z_i \tag{8}$$

The unbiased condition in Kriging defines that expectation of errors should be zero; it is showed in the equation (9).

$$E(z_o - \sum_{i=1}^{s} \lambda_i \beta\, z_i) = 0\,; \tag{9}$$

The λ values are obtained minimizing by least square the equation (10).

$$E(z_o - \sum_{i=1}^{s} \lambda_i \beta\, z_i)^2 = minimizing\,; \tag{10}$$

$$\begin{bmatrix} \Sigma_z & X \\ X' & 0 \end{bmatrix} \begin{bmatrix} \lambda \\ m \end{bmatrix} = \begin{bmatrix} c \\ x_0 \end{bmatrix} \tag{11}$$

In the equation (10) (Johnston et al., 2001) the sum of all λ values should be equal to 1. In this case, Lagrange multiplier m should be used.. X represents the order of the trend function. In the first order, X adds x,y coordinates to matrix (equation 11). In a second order trend would be needed to use all polynomial coefficient of the second order. Σ_z represents the theoretical semivariance matrix. c represents the values of unknown semivariance calculated through theoretical model. x_o defines the coordinates of the unknown point. Finally, the λ values are replaced in the equation (1) to get the unknown value z_0.

2.2.1.3 Radial Basis functions

This method works in a similar way as the Kriging interpolator, but without semivariograms. The basis of this method is centralized in the equation (12).

$$z_o = \sum_{i=1}^{s} w_i \varphi(r_i) + m \qquad (12)$$

Where, $\varphi(r_i)$ is the radial basis function, r_i is the distance from point p_0 to the i^{th}, the weights w_i and m which is the Lagrange multiplier. That information is organized on the matrix equation (13). Where Φ_z is the evaluation of all points in the function used[6]"

$$\begin{bmatrix} \Phi_z & 1 \\ 1' & 0 \end{bmatrix} \begin{bmatrix} \lambda \\ m \end{bmatrix} = \begin{bmatrix} \varphi \\ 1 \end{bmatrix} \qquad (13)$$

This method can be implemented with several functions such as Multiquadric, Inverse Multiquadric, Multilog, Thin Plate Spline, Natural Cubic Spline, Spline with Tension and Completely Regularized Spline Function[7] (Appendix B).

[6] http://www.spatialanalysisonline.com/output/html/Radialbasisandsplinefunctions.html

[7] http://www.spatialanalysisonline.com/output/html/Radialbasisandsplinefunctions.html

2.2.2 Cross Validation

The selection of a method of interpolation should be analyzed using the quality of the estimation through root mean square (RMS) equation (14) and the standardized RMS equation (15) for Kriging methods. The cross validation is executed at all the points in the dataset following the next steps (Chang, 2004):

a) A point is removed

b) The interpolation in the position of the eliminated point is calculated in order to estimate it.

c) The error is obtained by comparing the known value and the estimated value.

$$RMS = \sqrt{\frac{1}{n}\left(\sum_{i=1}^{n}(z_{known} - z_{estimated})^2\right)} \tag{14}$$

$$Standardized\ RMS = \sqrt{\frac{1}{n}\left(\sum_{i=1}^{n}{(z_{known} - z_{estimated})^2}/{s^2}\right)} \tag{15}$$

2.3 Cloud Computing

The Cloud Computing is not yet defined perfectly (Liu & Liu, 2010; Zhang et al., 2010a; Armbrust et al., 2009). According to Boss et al. (2007), it can be defined as a platform and an application. It is related with the quantity and configuration of the involved servers. Generally, it combines data storage, network infrastructure and security. Applications on the Cloud can be accessed using web services from anywhere. Grossman (2009) says that “Clouds or cluster of distributed computers provide on-demand resources and services over a network, usually the Internet, with the scale and reliability of a data center”, indicating that some typical applications like e-mail and social networks can be considered as Cloud applications. The fundamental idea of Cloud Computing is not new (Vouk, 2008; Grossman, 2009; Zhang et al, 2010b; Foster et al. 2008); it combines some grid computing attributes as scale, application oriented and services oriented. Xu (2010); Mikkilineni & Sarathy (2009) argued that current Cloud technology also shares similar characteristics with the evolution of the telecommunications infrastructure such as supporting new services and data sharing at large scales. Vaquero et al. (2009) presented more than 20 definitions of Cloud Computing which shows that a real definition is needed to evaluate its real benefits.

2.3.1 Overview

Cloud Computing is a complex combination of technologies, hardware, software, businesses, customers with some characteristics such as: user friendliness, virtualization, Internet centric, variety of resources, automatic adaptation, scalability, resource optimization, pay per use, ultra large-scale (thousands of servers), high reliability (fault tolerance), versatility (support different applications at the same time); high extendibility (grow dynamically); extremely inexpensive (Zhang et al., 2010a; Gong et al., 2010; Vaquero et al., 2009).

On the other hand, the adoption of Cloud Computing is limited by some obstacles, defined by Armbrust et al. (2009) as availability of a service, data Lock-in, data confidentiality, data transfer, scalable storage, scaling time, and software license. Table 2 describes each obstacle.

Obstacle	Effect	Who would be concerned?
Availability of a Service	Companies need to be sure about Quality of Service	Banks, Governments, large companies.
Data Lock-In	Difficulties to get data in distributed environments	All users.
Data Confidentiality and Audit ability	It is not possible to control where the information is stored and who manage the servers where information is located.	Governments, Large Companies
Data Transfer Bottlenecks	Accessibility problems when there are simultaneous user	Large companies, Governments
Scalable Storage	Problems in the definition of the database model (Relational Database or Blob schemas)	All Users
Bugs in Large-Scale Distributed Systems	Difficulties to model the environment of Cloud Computing	All Users
Scaling Quickly	Improving time of scaling without violating service level agreements	Companies, Governments
Software Licensing	Reduction of cost licenses	All Users

Table 2. Obstacles of Cloud Computing

Cloud Computing combines technologies for storing and distributing information using virtualization tools (Liu & Liu, 2010). The virtualization technology used in Cloud Computing is based on VMware, Xen and KVM. Although, there is not a specific programming model in Cloud Computing, the model MapReduce is increasing its adoption to process large datasets. Google, Amazon and Yahoo are using it to support huge datasets. Also, the BigTable technology is being used to manage huge datasets through redundancy mechanisms. On the other hand, security is an important concern on Cloud Computing. Both, government and companies require protocols of high security to put their information on the Cloud. The improvement of all security aspects involved on the Cloud to promote the adoption of this technology it is needed (Velte et al., 2010).

2.3.2 Types of Cloud Computing

Cloud Computing works based on the principles of a service-oriented architecture, that allows integrating and providing services. The term service is the common denominator between all types of Cloud Computing and it is related with the component used by vendor's network (Velte et al., 2010). Nowadays, there are different types of models of Cloud Computing that use the term XaaS to refer (Software, Platform, Hardware, etc.) to a Service. Although, it is possible to find other model such as: (Development, Database, and Desktop) as Service, Infrastructure as a Service, Business as a Service, Framework as a Service, Storage as Service, Organization as a Service (Rimal et al., 2010; Wu et al., 2010). These models share similar characteristics.

2.3.2.1 Infrastructure as a Service (IaaS)

This model of Cloud Computing is also called Hardware as a Service (HaaS), which provides the hardware that is required by customers. This architecture supplies resources as: CPU cycles, storage space, network equipment, and memory. The providers of IaaS also include tools for scaling down and up of resources, depending on users needs (Velte et al., 2010). Usually, the customer pays by the used resources.

2.3.2.2 Platform as a Service (PaaS)

The PaaS model provides the resources to deploy applications on Cloud Computing. This environment includes tools for designing, development, testing and hosting (Zhang et al., 2010a; Velte et al., 2010; Xu, 2010). With this model it is not necessary for client software to create new applications. For example, Google App Engine is configured to support applications of users that can be deployed automatically on the Cloud (Rimal et al., 2010). This platform provides all the resources that the application needs. On the other hand, the PaaS can be used to customize other type of software on the Cloud, but the developments created on a PaaS suffer problems to be moved between PaaSs.

2.3.2.3 Software as a Service (SaaS)

This model of Cloud Computing provides applications which do not require customer support. The updating of SaaS applications are done by providers. Usually, the customer should only pay for the time that the application is used. The SaaS applications are based on web applications and save cost licenses (Zhang et al., 2010a). They can be accessed from wherever, and they can support several customers at the same time (Rimal et al., 2010). This model saves money and provides better reliable applications. Volte et al., (2010) describes other benefits such as: more bandwidth, the applications can be customized easily, the applications will have better marketing, companies will need less IT staff, and the providers can configure security environments for each company.

2.3.3 Cloud Computing providers

2.3.3.1 Amazon Web Services (AWS)

The computation infrastructure of AWS is a changeable platform that provides different types of products such as computational infrastructure, database support, monitoring of services, management of messages and networking utilities. Some services are described below:

- Amazon Elastic Compute Cloud (EC2): This service provides an environment to create and manage instances, which refers to virtual servers with a variety of operating systems; they are called Amazon Machine Image (AMI). EC2 environment can be controlled by the web console or the web service API. This product is elastic due to its capacity of increasing or decreasing the number of instances[8]. The price of each instance depends on the running time, its location and its processing capacity; it can vary between $0.02 and $2.1 per hour.
- Elastic Block Store (EBS): store data independently of instances.
- Multiple Locations: It is possible to launch an instance in several locations.
- Elastic IP Address: The static IP is associated with the user account instead of a specific instance.
- Auto Scaling: This function allows increasing or reducing the number of instances depending on some predefined rules.
- Elastic Load Balancing: This tool distributes the requests among instances.
- VM Import: It is possible to import new virtual machine images to convert it on an AMI.
- Amazon Simple Storage Service (Amazon S3): (Amazon, 2010) argued that this product is designed to store information with a 99.999999% durability and 99.99% of availability. The redundancy is used to provide this level of service.
- Amazon CloudFront: This tool optimizes the transfer speeds among instances and end users.
- Amazon Simple Queue service: This service manages the messages between components in queues to prevent lost messages and improve the process of scalability.
- Amazon Relational Database Service (Amazon RDS): This service allows creating relational databases that support scalability and flexibility.

2.3.3.2 Google App Engine GAE

The configuration of GAE allows users to create web applications using languages as Python and Java. The infrastructure of GAE dynamically supplies the resources that the application needs; if an application increases its traffic, GAE scales the resources

[8] http://aws.amazon.com/ec2/

automatically to support it. In this way, the efficiency of the developers improves because they should not spend time solving infrastructure problems. GAE is designed to support Google products such as Google Docs, Calendar, Reader, etc. The GAE platform manages quotas and limits to publish applications. This limit allows for the conservation of the performance of the entire system. Quotas are related with the resources that can be used by users. The GAE has a limit of 30 seconds for over all requests. If a request has a longer duration, it is cancelled[9]

2.3.3.3 Windows Azure

The Windows Azure platform is focused on running and storing applications. It is a kind of PaaS, in which developers can deploy their application without thinking about infrastructure issues. The parts of Windows Azure are:

- Compute: The applications should be created using .NET Framework using languages as C#, Visual Basic, C++, Java, etc. The operating system is Window server.
- Storage: This platform support large objects, and traditional relational databases.
- Fabric controller: This part controls the jobs operation in the entire system.
- Content delivery network: Using the caching technique the Windows Azure increase the speed of data access.
- Connect: Windows Azure allows companies to interact with Cloud applications through independent applications, web applications and the SaaS implementations with the Microsoft technology (Chappell, 2009).

2.3.4 GIS in Cloud Computing

The GIS technology manages large datasets and requires high computational resources. Jinnan & Sheng (2010) argued that GIS on Cloud can improve the capacity of GI storage and processing. Cloud Computing can supply these needs and adds other useful features such as: better GI distribution, high computational power, accessibility anywhere, etc. Singh & Wen (2010) argued that it is possible to process terabytes of information harmonizing the price of services and the capacity

[9] http://code.google.com/appengine/docs/whatisgoogleappengine.html

of processing on Cloud Computing. At this moment, applications that provide GI such as Google maps work with Cloud technology to support thousands of users (Velte et al., 2010), and GIS applications as Mapinfo and ArcGIS Server provides capabilities to process and manage GI on Cloud Computing environment. Also, other projects have demonstrated to be useful for the distribution of GI on Cloud Computing e.g., Blower (2010) evaluated the feasibility of WMSs on GAE, although there were some limitations due to restrictions of GAE. The results demonstrated that it was possible to include geographical characteristics on GAE. On the other hand, the management of large GI datasets in Internet requires new types of indexing; Cary et al. (2010) implemented a new index through Hadoop technology over a dataset of 110-million property parcels in a private Cloud.

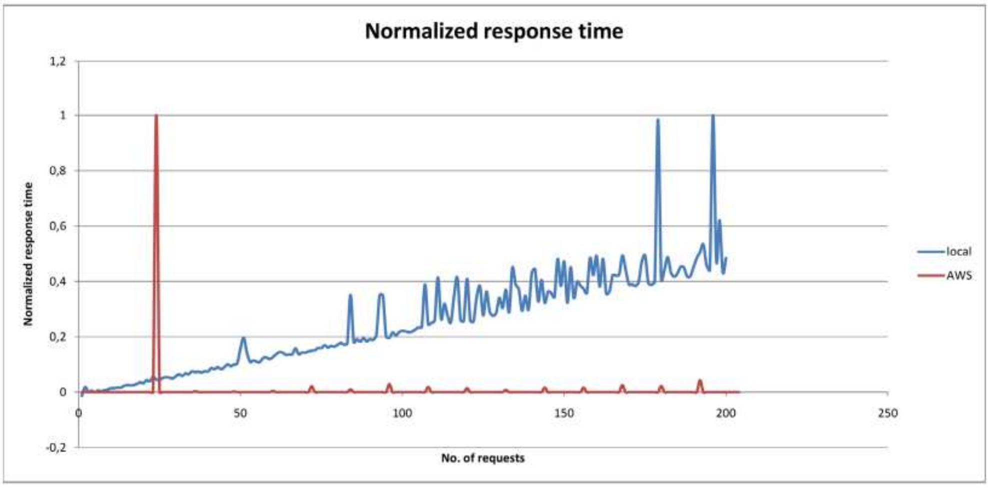


Figure 2. Response WPS on Cloud Computing (Schäffer et al., 2010)

The Cloud Computing can improve the performance of a SDI application when the number of users increment. (Schäffer et al., 2010) showed that a WPS service can be scaled on AWS infrastructure without damaging its performance. Figure 2, shows that the WPS performance on AWS is almost constant although the number of requests increases.

2.3.5 Relationship between Grid computing and Cloud Computing

The Grid computing is defined as a set of computers dedicate to solve a problem in parallel (Velte et al., 2010), but this system has similar features to Cloud Computing. Table 3 shows a comparison of Cloud and Grid computing, where aspects as architecture, programming model, resource management, and service negotiation are common in both systems.

The differences between both systems do not imply that they can work together. Platforms as GAE and AWS are using parallel paradigms like MapReduce that previously were exclusive to Grid computing. Zhang et al. (2010b) stated that "*Now the dream of grid computing will be realized by Cloud Computing. It will be a great event in the IT history*". The integration of Cloud and grid computing nowadays is evident, some databases are using grid paradigms to improve the management of large dataset on Cloud, and several middleware from Grid computing are being using on Cloud Computing to provide more computational power. Figure 3 shows the integration between Cloud and grid computing.

Characteristic	Grids	Clouds
Node operating system	Dominated by Unix	Virtual Machines
Service negotiation	Service Level Agreement (SLA)	SLA
Resource management	Distributed	Centralized, Distributed
Allocation	Decentralized	Centralized, Decentralized
Value-added	Limited	High Potential
Users	Few	Many
Cost	High – fix	Cheap - Variable
Architecture	Application and Service Oriented	Service Oriented
Programming model	Parallelization Paradigms, MapReduce	MapReduce
Security	Complex model	Simple model

Table 3. Comparison between Cloud and Grid computing

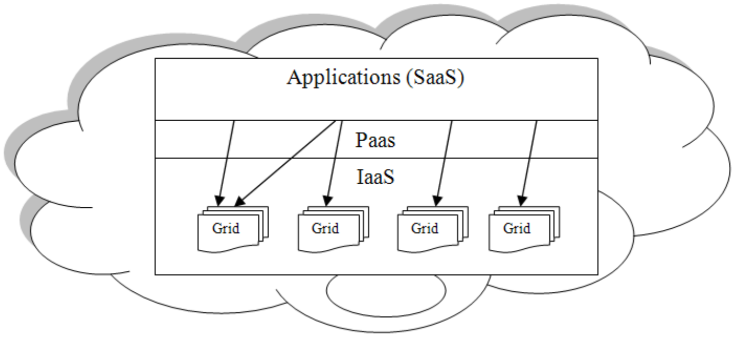


Figure 3. Integration between Grid and Cloud Computing

3. Resources used

3.1 Description of software and hardware used

The software and hardware used in this project can be divided on two stages:

- Programming phase
- Testing phase

The list of software products used in the programming phase is:
Java SDK 1.6.0.20, Eclipse Galileo 3.5.2, Operating System Ubuntu – Linux 10.04 LTS Lucid Lynx, GridGain 2.1.1, 52North WPS Framework RC6, Tomcat 6 and Geoserver 2.2. Some WPS clients as: 52North WPS OpenLayer client, and OpenJump 1.4.0.

The hardware used:
1 laptop AMD Athlon™ X2 Dual-Core QL-64, memory 3.0 GiB, hardisk 250 GiB.

In this phase, the configuration of a second computer is required with the software: Java SDK 1.6.0.22, Operating System Windows XP, and GridGain 2.1.1. The specification of the computer is: Intel® core™ 2 Duo, memory 3.0 GiB. The connection was through ad hoc network. The GIS program used is ArcGIS 9.3.

In the testing phase the software products used are referred as AMI micro instances in AWS. A master node contains the following software products: Operating system fedora, Tomcat 6, Java SDK 1.6.0.17, GridGain 2.1.1, Geoserver 2.2, 52North WPS Framework RC6 and 52North WPS OpenLayer client. In addition, 9 nodes with the following software: Operating system Fedora, Java SDK 1.6.0.17 and GridGain 2.1.1. All instances have been created in paravirtualization mode and their hardware simulates: one core with 613 MB.

3.2 Description of data used

The Geostatistics methods are usually applied over events, samples or other variables with a continuous behavior such as temperature or elevation. According to this assumption the datasets selected for testing in this project are: The maximum temperature in the continental part of Spain and an elevation dataset.

3.2.1 The maximum daily temperature dataset

The maximum daily temperature is a meteorological variable that is captured by the weather stations. This variable is essential for calculating some agricultural parameters such as growing degree days or heating degree days[10]. The dataset used in this project is published by The Meteorological Agency of Spain[11] (AEMET). The dataset contains the information of 569 stations without including stations located on Canarias and Africa (Figure 4). The dataset date is January 23, 2011.

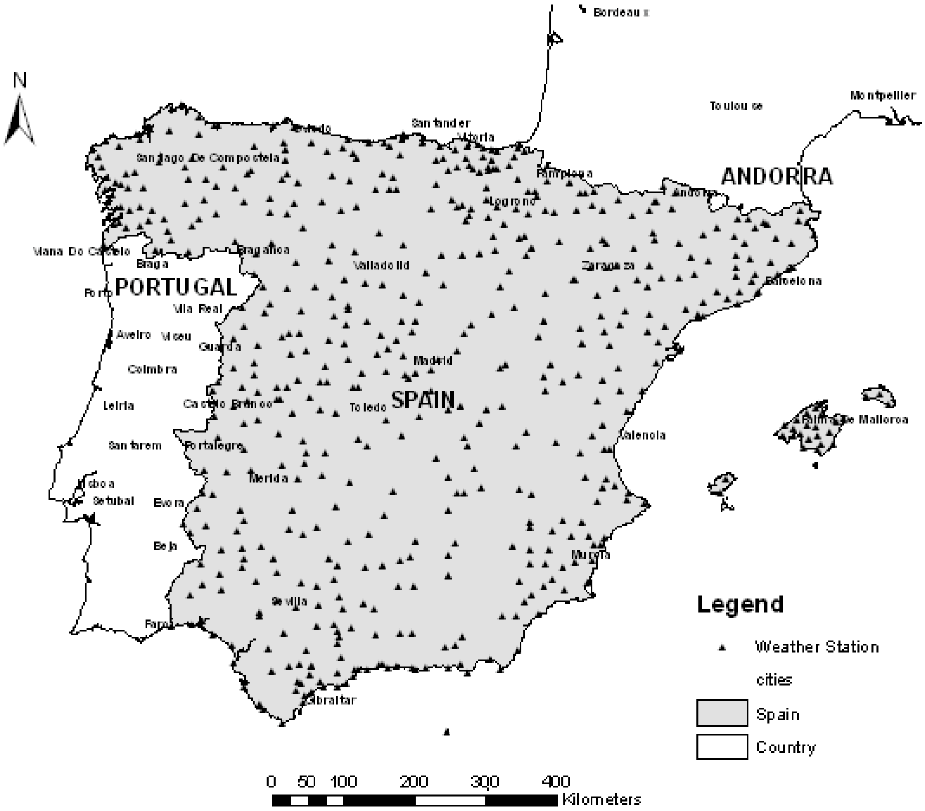
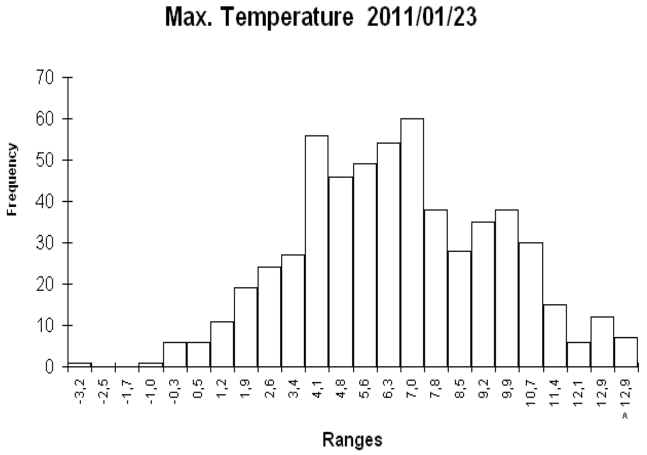


Figure 4. Distribution of weather stations

The dataset describes a Gaussian distribution with a small positive skewness that indicates a concentration toward high values (Figure 5).

[10] http://www.gov.ns.ca/agri/ci/weather/reports/definitions.asp

[11] http://www.aemet.es/es/servidor-datos/acceso-datos/listado-contenidos/detalles/datos_observacion



Max Temp.	
Mean	6.2
Median	6.1
Mode	5.7
Standard Deviation	3.1
Variance	9.3
Kurtosis	-0.4
Skewness	0.1
Range	16.8
Minimum	-3.2
Maximum	13.6
Sum	3524.4
Count	569

Figure 5. Statistical distribution of the maximum temperature dataset

3.2.2 Elevation dataset

The elevation samples are commonly used to generate digital surfaces or Digital Elevation Models DEMs. Nevertheless, this work uses a DEM to create two datasets of 1000 and 10.000 samples. The DEM is published by OpenTopo[12] and created in 2008, with a resolution of 0.5 meters, covers an area of 400 hectares and it is based on LIDAR technology (Figure 6). The coordinate system is Universal Traversal Mercator (UTM) region 12 North, with datum WGS84.

The statistical distribution of the DEM describes a bimodal shape (Figure 7), and it is non-normal. The dataset collected describes a non-normal distribution (Figure 8) similar to the distribution of DEM. The mean between DEM and samples differs in 1.21 units, and standard deviation in 0.26 units.

[12] http://opentopo.sdsc.edu/gridsphere/gridsphere?cid=datasets

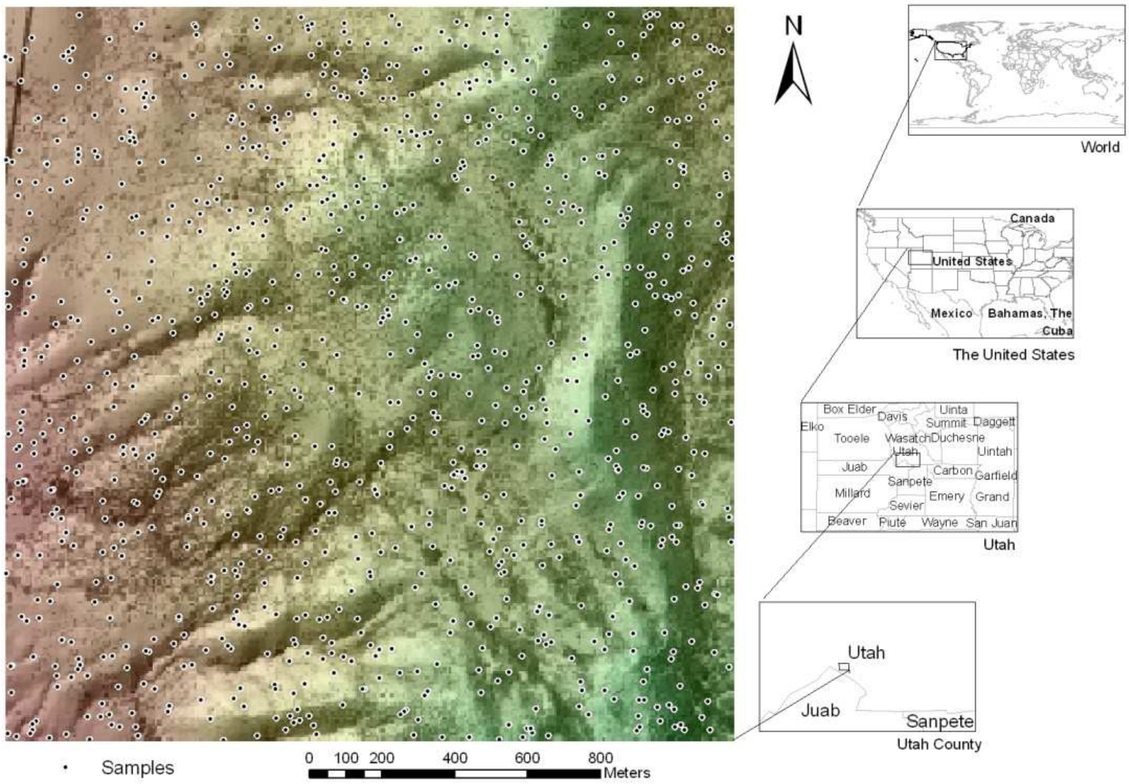
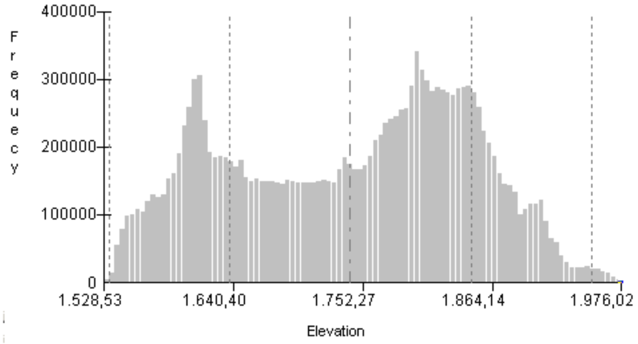
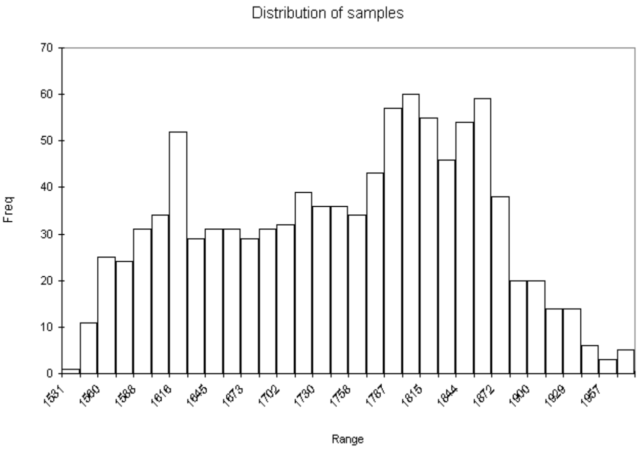


Figure 6. Distribution elevation samples in the dataset.



DEM features	
Mean	1741.41
Std. Dev	104.36
Sum	27x10^9
Minimum	1528.53
Maximum	1976.02
Count pixel	16x10^6
Resolution	0.5

Figure 7. Distribution and statistics about DEM used



Z	
Mean	1740.2
Median	1756.3
Mode	1806.8
Standard Deviation	104.1
Variance	10833.9
Kurtosis	-1.0
Skewness	-0.2
Range	440.2
Minimum	1531.1
Maximum	1971.3
Sum	1740187.1
Count	1000

Figure 8. Statistics of samples used

4. Geostatistical methods library

This chapter presents the key issues used to create a Geostatistical library that implements four interpolator methods such as Ordinary Kriging, Universal Kriging, IDW and RBF. Also, the chapter describes the mechanism used to determine the best parameters of each method.

This Geostatistical library will be used to evaluate the performance of WPS on the Cloud. Although, some Open Sources applications and libraries e.g., Sextante[13], Gslib[14], Gstat[15], R[16], etc., have geostatistical capabilities, only the Sextante library works with Java technology which it is the technology used by the WPS framework. Besides, the 52North Framework includes the Sextante Java libraries by default. The Sextante library supports several geographic functions including some interpolator methods, but it is not focused on Geostatistical problems; otherwise, other libraries to work are needed. In the process of parallelization these libraries should be also sent to each node. This thesis work prefers to develop a new simple library with parallelization capabilities to evaluate the parallelization of WPS services on the Cloud. This option, avoid sending libraries that will not be used in the nodes and allows for the control of all parameters of the Geostatistical methods.

[13] http://forge.osor.eu/projects/sextante/
[14] http://www.gslib.com/
[15] http://www.gstat.org/whatsnew.html
[16] http://cran.r-project.org/index.html

4.1 Interpolator requirements

For the creation of a Geostatistical library the similarity between each method of interpolation needs to be determined; these similarities allow for defining some especial requirement that the library need (Table 4).

Although, the Ordinary and Universal Kriging methods can work as global interpolators, in this research they are managed as local interpolators to avoid inverting huge matrices in the process of interpolation. All methods use the points around to execute the interpolation. However, the methods Ordinary Kriging, Universal Kriging and RBF manage matrices in the process to determine the weights to interpolate. The Kriging methods use standardized RMS to define the best parameters.

Method	Matrix management	Sub models	Selection of points around	Fitting sub model	Selection best parameters
Kriging	x	x	x	x	RMS, Std RMS
Kriging Univ.	x	x	x	x	RMS, Std RMS
IDW	-	-	x	-	RMS
RBF	x	x	x	-	RMS

Table 4. Similar aspects between each method

4.2 Design Geostatistical library

Using the definition and requirement of each Geostatistical method, four packages need to be created to manage the requirements of the library (Figure 9).

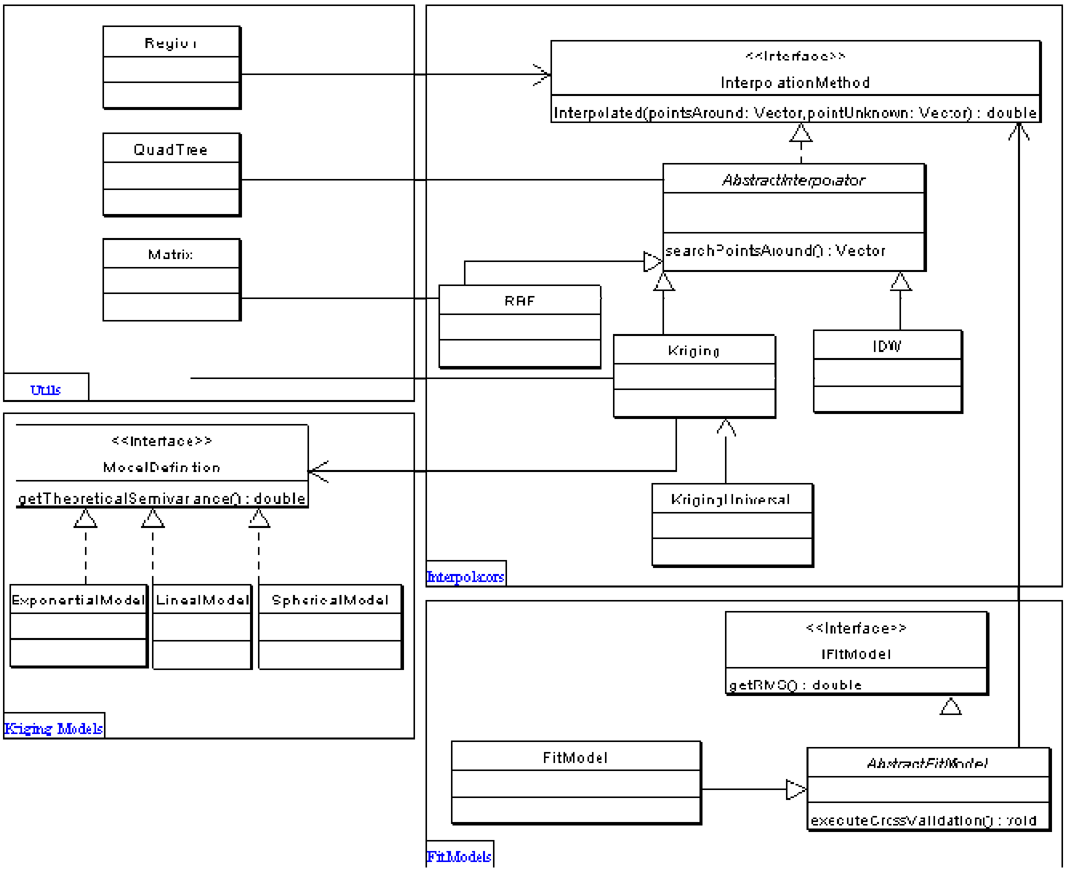


Figure 9. Geostatistical classes diagram

The first package called Interpolators defines a class for each method, one abstract class and one interface. The most relevant operation defined in the interface is Interpolate. This operation receives the closest points around a location and the point of this location. The second package called Fitting finds the best parameters for each method using cross validation method. The third package called Kriging Models defines some models that can be used in the interpolation; in this case the models used are Lineal, Spherical, and Exponential. The fourth package called Utils manage some classes that support the interpolation over a region, and contains some special classes as Matrix, to invert matrices, and Quatree, to create an index to improve the selection of point around.

4.3 Determining the best parameter for each method

Each interpolator method needs certain initial parameters to execute the interpolation e.g., IDW needs the power value to run or Kriging needs to use a specific model to interpolate.

The evaluation of an interpolation can be done by cross validation and validation methodology (Chang, 2004). This project uses the cross validation method to determine the best interpolator method through the use of RMS and standardized RMS in the case of Kriging. Usually, all parameters can be changed by the user, but there are some parameters that are pre-defined by the application e.g., parameters in ArcGIS (Table 5).

Method	by user	by default	by software
Kriging	Number of points around to search, distance of searching, theoretical model, anisotropy (angle), trend	lag=12	Range, sill, nugget
IDW	Number of points around to search, distance of searching, theoretical model	Power=2	
RBF	Number of points around to search, distance of searching model		Smooth factor

Table 5. List parameters used by ArcGIS

The Geostatistical library evaluates a range of values for each parameter and calculate the cross validation for obtaining the best option. The application returns the parameters with the lowest RMS for methods IDW and RBF, and in the case of Kriging it returns the parameters based on two criteria, the lower RMS and closest standardized RMS to 1. The parameters used in this implementation vary according with a pre-defined range that the user can change.

The IDW method uses the following ranges: Number of neighborhoods from 5 to 12, and the power value change from 0.1 to 10 at steps of size 0.1.

The Kriging methods use the following ranges: N Number of neighborhoods from 5 to 12, number of lags (7-15), lag's length (dividing the maximum length in four parts) and the models: Lineal, Spherical and Exponential. The RBF method uses the following ranges: Number of neighborhoods from 5 to 12, the smooth factor from 0.1 to 0.5 at steps of size 0.1 and seven models (Appendix B).

4.4 Implementation of Geostatistical library

The Geostatistical library is developed in Java using the Eclipse platform following the UML diagram (Figure 9). This library has added two generic classes: QuadTree[17] and Matrix[18] (Sedgewick & Wayne, 2010) which are needed to execute some functions in the process of interpolation. A general approach about the interpolation process is presented in the figure 10. This function receives an array with the points that have an influence over the unknown location. Both parameters are used to get the weights. Finally, each weight is related with a point, and then the interpolated value is the addition of multiplication between each weight and its Z value.

Pseudo-code: Generic Interpolation procedure

```
Function Interpolate (points around, unknown location) as Double
    Weigths = GetWeight (points around, unknown location)
    Value interpolated = 0
    For  i to number of points around
        Value interpolated + = Weigths(i) * Z(points around (i))
    Next
    Return Value interpolated
End Function
```

Figure 10. Pseudo-code Generic interpolation procedure

The function GetWeight is particular for each interpolator method. The Ordinary Kriging uses the equation (7), the Universal Kriging equation (11), the RFB equation

[17] http://www.cs.princeton.edu/algs4/92search/QuadTree.java

[18] http://introcs.cs.princeton.edu/95linear/Matrix.java.html

(13) and IDW the equation (3). Figure 11 illustrates the GetWeight function of the Ordinary Kriging. The evaluation of this library is presented in the results section.

```
Pseudo-code : GetWeight  Ordinary Kriging
Model =Theoretical Model (range, sill, nugget)
 Function GetWeight (points around, unknown location)  as Matrix
     Matrix semivar =Matrix [ num points around + 1,  num points around + 1]
     Vector to unknown location =   Matrix [num points around + 1, 1]
    DefineMatrixStructure (Matrix semivar) // Add multiplier Lagrange values
     For i  to num points around
          For  j to num of points around
            Matrix semivar[i , j] =Model.GetValue( Distance point (i , j))
         Next j
           Vector to unknown location [i, 0]  =
                         Model.GetValue( Distance point (i , unknown location))
      Next i
      Matrix inverse = Matrix.Inverse(Matrix semivar)
      Return Matrix inverse X Vector to unknown location
   End Function
```

Figure 11. Pseudo-code GetWeight Ordinary Kriging.

The function DefineMatrixStructure fills the last row and column in the matrix with values 1. The Model.GetValue returns the semivariance according to the theoretical model selected. The procedure returns a vector with the number of points + 1.

5. Geostatistical library on the WPS framework

The 52north WPS framework is used in this thesis to support the creation of geoprocessing services. The Geostatistical library is added and configured in the framework in order to create geoprocessing services.

In the creation of a new service e.g., Interpolation, it extends the AbstractObsevableAlgorithm Class (Figure 12), this abstract class manages the association with the WPS description.

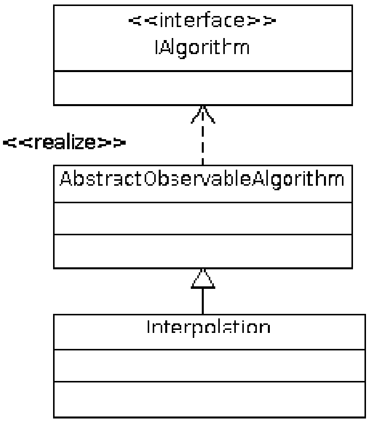


Figure 12. Addition a new algorithm in the 52North WPS framework

Each service has a description file with the WPS features, and it should be added to the repository through the configuration file called wpsConfig[19]. On the other hand, the new algorithms related with this Geostatistical library uses the libraries of Geotools[20] and OpenGis API[21] to support the output in the GeoTiff format . The result of the interpolation is a surface which is stored in Geoserver. The 52North WPS framework sends the information through a REST service provided by the Geoserver.

5.1 Designing the parallelization profile of interpolation methods

The parallelization process depends on the type of infrastructure used in the implementation and the level of parallelization needed. For example, it is possible to parallelize the matrix inverse process which is included in some interpolation methods. However, this could require high bandwidth and low latency between nodes involved in the process. The nodes allow data managing and processing.

[19] wpsConfig: This the configuration file of the 52North WPS Framework
[20] http://www.geotools.org/
[21] http://www.geoapi.org/

This profile describes where the interpolation process executes the parallelization and how the jobs are distributed. Several parallel interpolation algorithms have been suggested by Strzelczyk & Porzycka (2010); Pesquer-Mayos (2008). Figure 13 some techniques to divide the process of interpolation are illustrated.

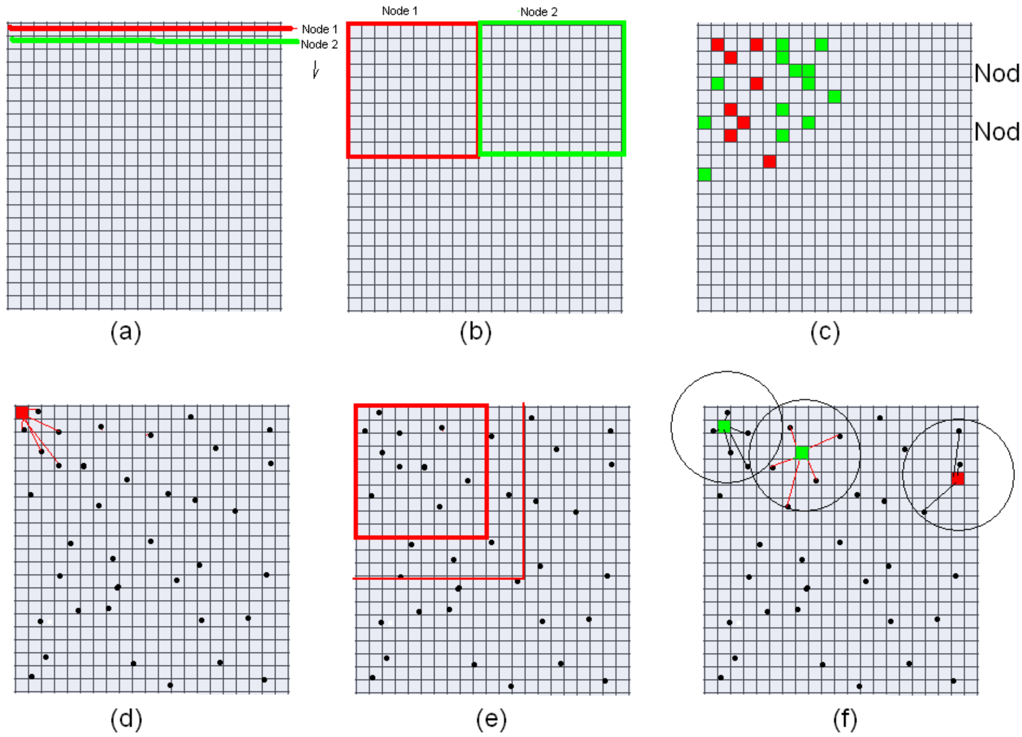


Figure 13. Techniques used to divide task in the interpolation.

The interest area is divided into cells or pixels that depend on the resolution or the selected number of columns and rows. The process of interpolation is executed in the location of each cell. In this step, the process can be distributed and executed among nodes in parallel. In the figure 13a, each row is assigned to a node that should execute the interpolation in each cell. The second option is to divide the area in sub regions (Figure 13b) to be sent to each node. Another option is to group the cells depending on density of points around. Figure 13c illustrates how some pixels with the same density are sent to each node.

In the process of job distribution the original point has to be sent to each node. In figure 13d, each node contains the information of the surrounding points that should be sent to the node. In the sub division technique (figure 13e) the points of an external region are added. The buffer size is based on mean distance between points are used. Figure 13f shows how the density of each pixel is calculated.

5.2 Adding parallel characteristics in the Geostatistical library

Using the parallelization profile of the previous section, some operations need to be added to the model in order to parallelize the library (Figure 14).

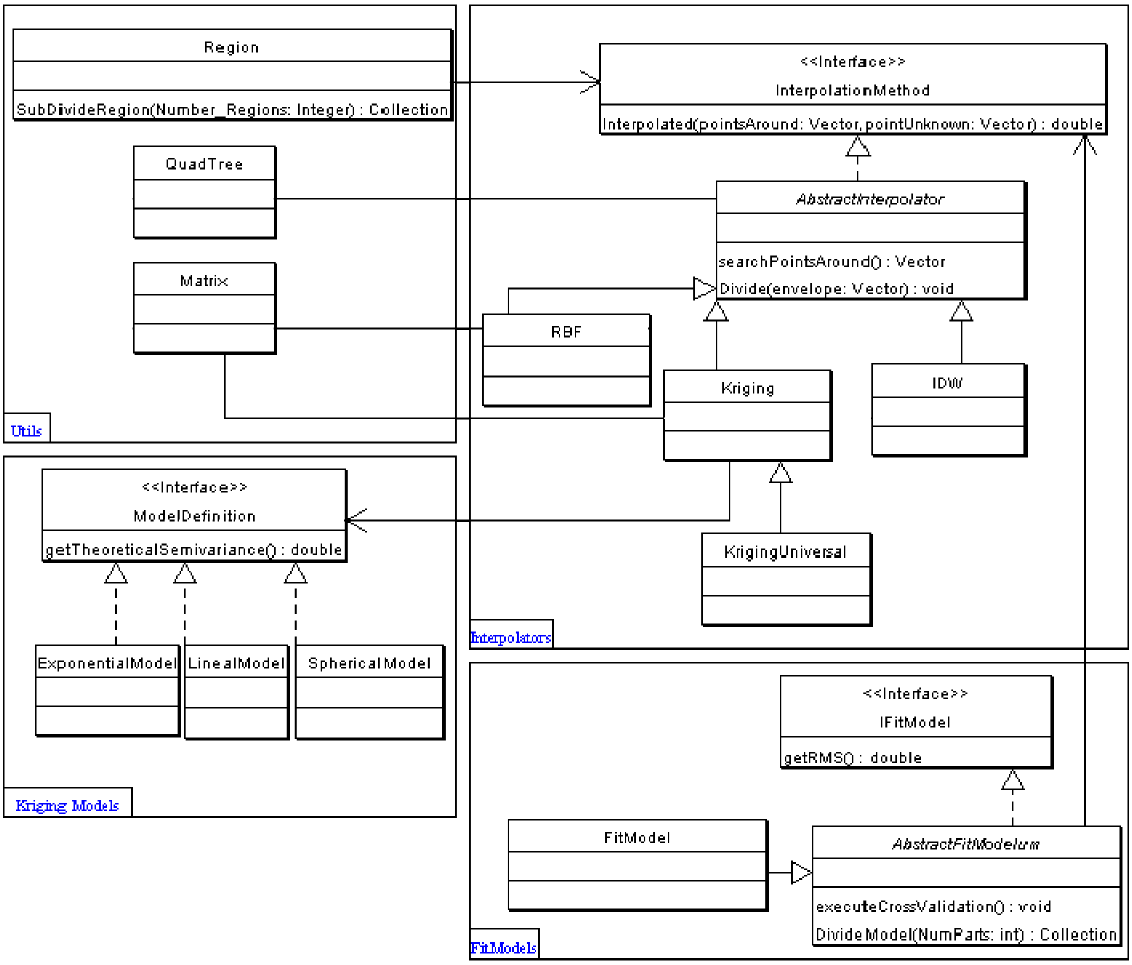
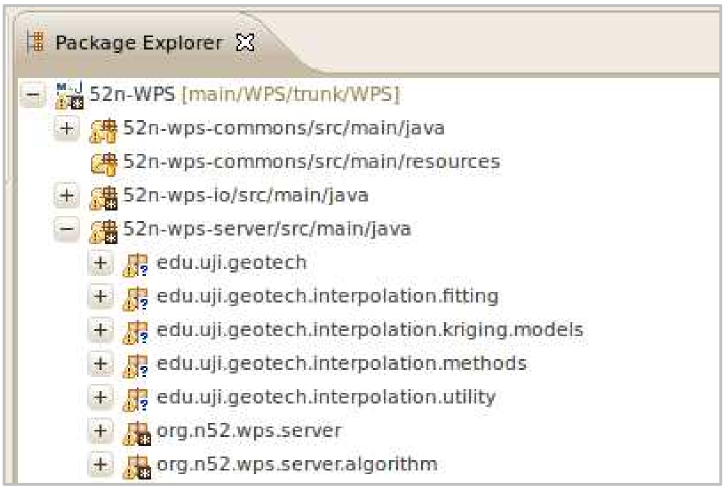

Region
SubDivideRegion(Number_Regions: Integer) : Collection
QuadTree
Matrix
Utils
<<Interface>>
InterpolationMethod
Interpolated(pointsAround: Vector, pointUnknown: Vector) : double
AbstractInterpolator
searchPointsAround() : Vector
Divide(envelope: Vector) : void
RBF
Kriging
IDW
KrigingUniversal
Interpolators
<<Interface>>
ModelDefinition
getTheoreticalSemivariance() : double
ExponentialModel
LinealModel
SphericalModel
Kriging Models
<<Interface>>
IFitModel
getRMS() : double
FitModel
AbstractFitModelum
executeCrossValidation() : void
DivideModel(NumParts: int) : Collection
FitModels

Figure 14. Geostatistical classes diagram with parallel capabilities.

The Region Class is in charge of splitting the area of interest into sub regions. The AbstractInterpolator Class has a new function that divides the dataset of points depending on the created region. In AbstractFitModel Class, an operation that divides the iterations needed to determine the best parameters is added. The library is added to the WPS framework (Figure 15) and allows for the creation of geoprocessing services that can be parallelized.



```
52n-WPS [main/WPS/trunk/WPS]
  52n-wps-commons/src/main/java
  52n-wps-commons/src/main/resources
  52n-wps-io/src/main/java
  52n-wps-server/src/main/java
    edu.uji.geotech
    edu.uji.geotech.interpolation.fitting
    edu.uji.geotech.interpolation.kriging.models
    edu.uji.geotech.interpolation.methods
    edu.uji.geotech.interpolation.utility
    org.n52.wps.server
    org.n52.wps.server.algorithm
```

Figure 15. Geostatistical library in the 52North WPS framework

5.3 Configuration of parallelization environment on the framework

The 52North WPS framework has two extensions to manage processes on Grid: GridGain and UNICORE. This project works with the extension GridGain to distribute processes and data on parallel (figure 16).

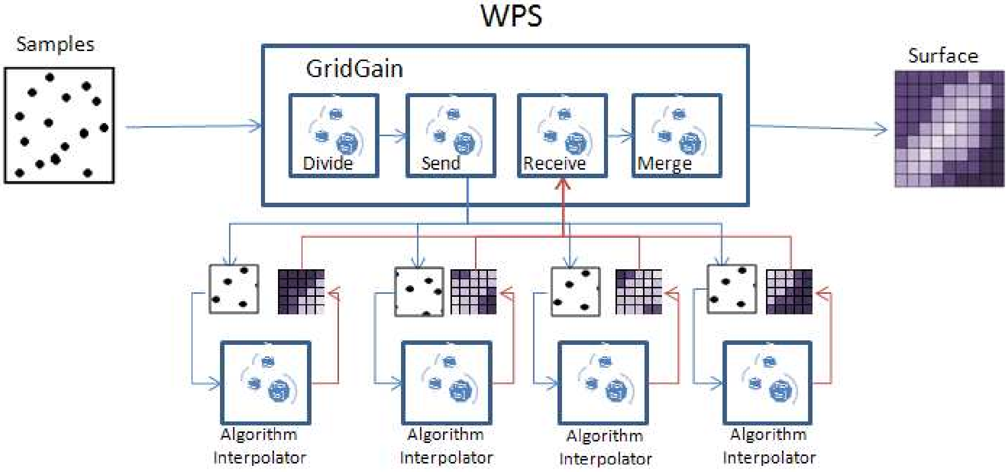


Figure 16. WPS with GridGain approach

This 52North WPS GridGain extension adds the libraries of GridGain for supporting its functionalities. Figure 17 illustrates the classes diagram involved in the publishing of a WPS service with Grid capabilities. The GridGainInterpolator Class splits the input data and merges the result of all nodes. Also, it configures the WPS service using the properties and functions of the Class AbstractGridAlgorithm. GridGain sends the data and the Interpolate algorithm to each node. The service

GridGainInterpolator should be added to the repository of algorithms of GridGain in the wpsConfig.

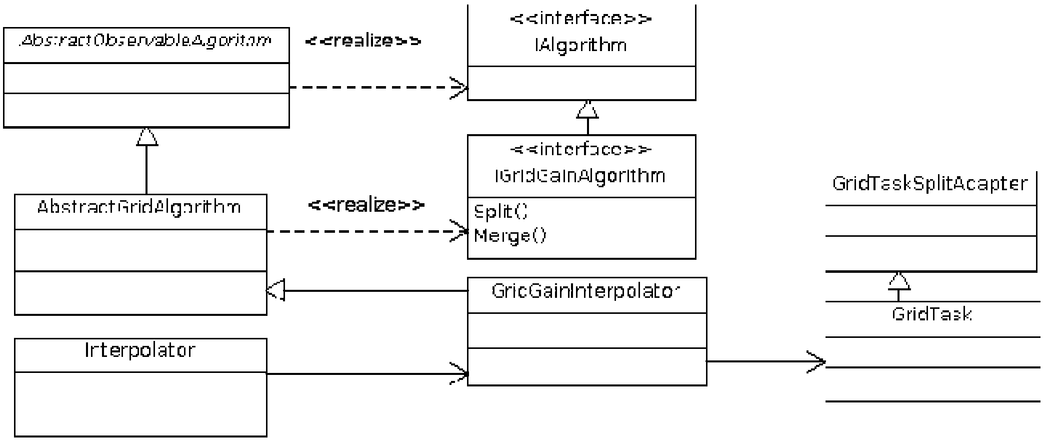


Figure 17. Extension of GridGain in the 52North WPS framework

The WPS service configured with GridGain follows these steps:

1. The service receives the data
2. One GridGain master node is started
3. The master node establishes a communication with other nodes
4. The number of parts in which the data will be divided is defined
5. The data are split
6. The algorithms and the data are sent to each node
7. Each node receives the algorithm and uses it to process the data
8. Each node return a result to the master node
9. The master node merges the processed data
10. The merged data is returned to the WPS service
11. The GridGain master node is stopped.
12. The WPS return the processed information

In this configuration, each request generates a new node and takes a while until its activation. This project suggests a new approach in the implementation of GridGain in the 52North Framework (Figure 18).

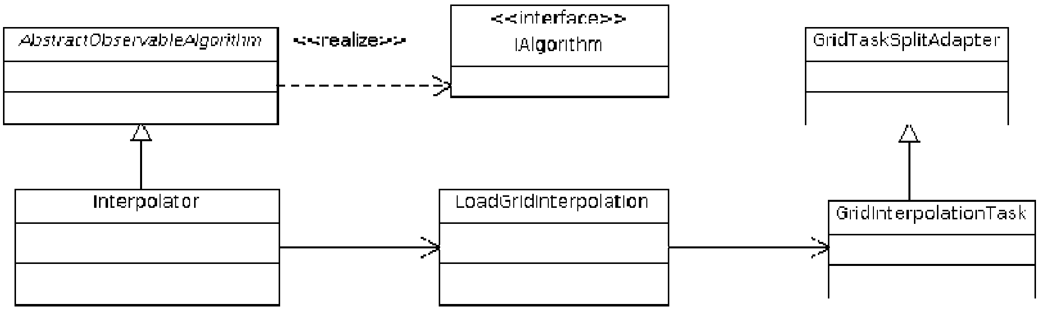
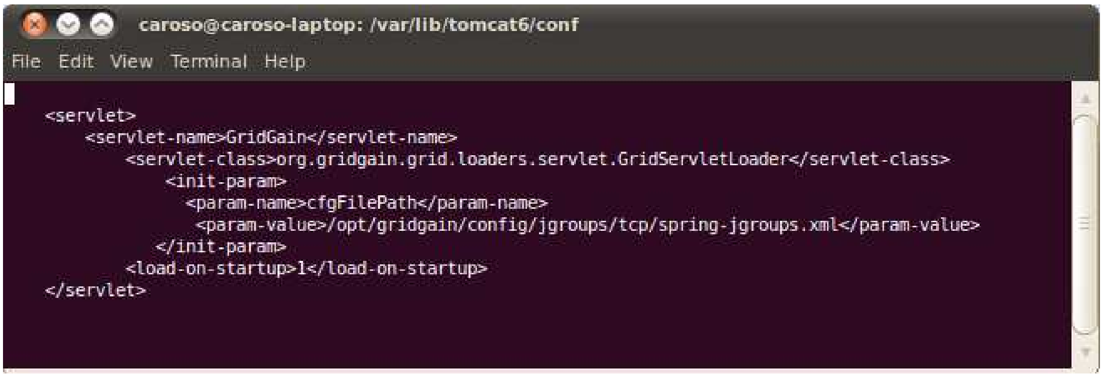


Figure 18. New GridGain approach in the 52North WPS framework

The implementation of this new approach follows these steps:

1. The service receives the data
2. A master node is searched
3. The master node establishes a communication with other nodes
4. The number of parts in which the data will be divided is defined
5. The data are split
6. The algorithms and the data are sent to each node
7. Each node receives the algorithm and uses it to process the data
8. Each node return a result to the master node
9. The master node merges the processed data
10. The merged data is returned to the WPS service
11. The WPS return the processed information

This approach uses an external master node to support the requests of the WPS services. This node is initialized adding some parameters in the Tomcat file web.xml (Figure 19).


caroso@caroso-laptop: /var/lib/tomcat6/conf
File Edit View Terminal Help

```
<servlet>
    <servlet-name>GridGain</servlet-name>
        <servlet-class>org.gridgain.grid.loaders.servlet.GridServletLoader</servlet-class>
            <init-param>
              <param-name>cfgFilePath</param-name>
               <param-value>/opt/gridgain/config/jgroups/tcp/spring-jgroups.xml</param-value>
           </init-param>
        <load-on-startup>1</load-on-startup>
</servlet>
```

Figure 19. Starting a GridGain node in Tomcat

5.4 Defining processes in the framework

The functions of the Geostatistical library are published as WPS services, in which the parameters and type of data supported by each function are defined. The WPS services with parallel functionalities should be configured with the GridGain capabilities. The lists of implemented services are:

General Cross Validation

This service allows for estimating the best parameters needed for a specific method of interpolation. The description of the service is found in the Appendix C.

- Inputs:
  1. Data: WFS with GML and SHP-ZIP format
  2. Field: Contain the attribute to do the interpolation
  3. Method: Ordinary Kriging, Universal Kriging, IDW, RBF
- Outputs:
  1. RMS: Error of the best method in the cross validation
  2. StdRMS: Standardized Error
  3. Correlation coefficient
  4. Parameters: Parameters found.
  5. Cross Validation Graph: URL with cross validation graph
  6. Fitting graph: Show the error behavior according to the method selected
  7. Iteration: URL with a log file with the iterations summary.

Interpolation

This method executes the interpolation according to the method selected. The description of the service is found (Appendix D):

- Inputs:
  1. Data: WFS with GML and SHP-ZIP format
  2. Field: Contain the attribute to do interpolation
  3. Method: This input receives a string with the method and parameters for executing the interpolation
- Outputs:
  1. Result: WMS with reference to coverage on Geoserver
  2. Duration: process duration

5.5 WPS client

The 52North OpenLayer WPS client is used to test the processes created, although some modification have been included into the Javascript client to support the GML 2.0 schema and the reference to one WMS service. The services created are not running correctly in this software due to problems in the WFS layer processing. When a WFS layer is loaded in the OpenJump and then used by the WPS extension, the extension does not recognize the attributes of the WFS Layer.

6. Implementing the WPS on the Cloud

The type of Cloud Computing platform needed for deployment of the WPS framework requires the support of Java libraries. The GAE platform does not support some libraries needed for deploying the parallel WPS.

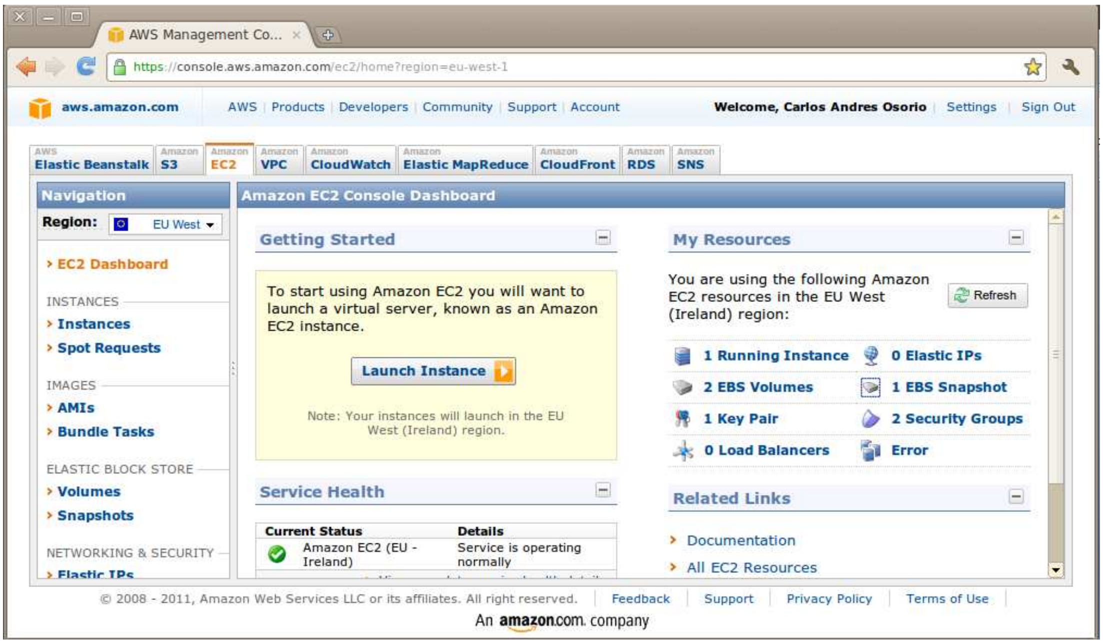
On the other hand, the AWS platform allows for the creation and configuration of servers with the requirements needed by the WPS framework (Baranski et al., 2010). This research uses the AWS platform for deploying the WPS framework on Cloud. On November 1, 2010 the AWS released a free account that provides a micro instance for one year without any cost. This work uses this account to evaluate the performance of the WPS in the platform.

6.1 Cloud environment configuration in the AWS platform

The AWS requires creating a new account for accessing its resources. All resources used are administrated by this account, and its principal component is the console (Figure 20). In the process of creating of an account the security credentials have to be defined which allows it to be accessed through secure REST or using AWS service API.

This research work uses the credentials to control the servers created in the platform through Secure Shell (SSH), Secure copy (SCP) and the AWS API. The instances can be accessed through the following command lines:

- ssh -i credential.pem ec2-user@ amazon.server
- scp -i credential.pem file ec2-user@amazon.server:/home/ec2-user/



AWS Management Co...
https://console.aws.amazon.com/ec2/home?region=eu-west-1
aws.amazon.com
AWS | Products | Developers | Community | Support | Account
Welcome, Carlos Andres Osorio | Settings | Sign Out
AWS Elastic Beanstalk | Amazon S3 | Amazon EC2 | Amazon VPC | Amazon CloudWatch | Amazon Elastic MapReduce | Amazon CloudFront | Amazon RDS | Amazon SNS
Navigation
Region: EU West
EC2 Dashboard
INSTANCES
Instances
Spot Requests
IMAGES
AMIs
Bundle Tasks
ELASTIC BLOCK STORE
Volumes
Snapshots
NETWORKING & SECURITY
Elastic IPs
Amazon EC2 Console Dashboard
Getting Started
Launch Instance
Service Health
My Resources
Refresh
Related Links
Documentation
All EC2 Resources
Feedback | Support | Privacy Policy | Terms of Use
An amazon.com. company

To start using Amazon EC2 you will want to launch a virtual server, known as an Amazon EC2 instance.

Note: Your instances will launch in the EU West (Ireland) region.

You are using the following Amazon EC2 resources in the EU West (Ireland) region:

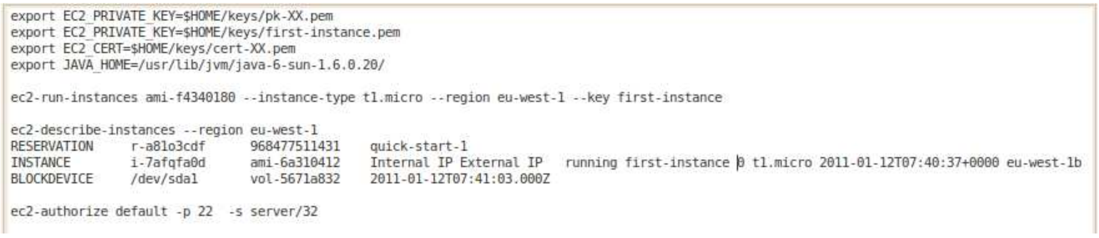
1 Running Instance	0 Elastic IPs
2 EBS Volumes	1 EBS Snapshot
1 Key Pair	2 Security Groups
0 Load Balancers	Error

Current Status	Details
Amazon EC2 (EU - Ireland)	Service is operating normally

© 2008 - 2011, Amazon Web Services LLC or its affiliates. All right reserved.

Figure 20. Console of the platform AWS

The credentials are needed for creating instances using command lines. The AWS API controls the whole environment of the AWS platform. In figure 21 the steps to configure the AWS API are shown as well as some commands used to create, to describe, and to alter the port permission in an instance.



```
export EC2_PRIVATE_KEY=$HOME/keys/pk-XX.pem
export EC2_PRIVATE_KEY=$HOME/keys/first-instance.pem
export EC2_CERT=$HOME/keys/cert-XX.pem
export JAVA_HOME=/usr/lib/jvm/java-6-sun-1.6.0.20/

ec2-run-instances ami-f4340180 --instance-type t1.micro --region eu-west-1 --key first-instance

ec2-describe-instances --region eu-west-1
RESERVATION     r-a81o3cdf      968477511431    quick-start-1
INSTANCE        i-7afqfa0d      ami-6a310412    Internal IP External IP   running first-instance 0 t1.micro 2011-01-12T07:40:37+0000 eu-west-1b
BLOCKDEVICE     /dev/sda1       vol-5671a832    2011-01-12T07:41:03.000Z

ec2-authorize default -p 22  -s server/32
```

Figure 21. Configuration AWS API

6.2 Addition of WPS on the Cloud

This project requires creating a server with the capabilities to support the deployment of the 52North WPS framework with the Geostatistical library and with the extension GridGain activated. Amazon AWS has approximately 2433 public instances[22] that can be used for creating an instance; this project uses a standard micro instance: *ami-6a31041e* with the kernel aki-*4deec4c43*. In this micro instance the following programs are installed: Java JDK, Tomcat 6, and GridGain. The 52North WPS, Geoserver and OpenLayer WPS client are deployed in Tomcat. In addition, the configuration of these programs is presented:

- export GRIDGAIN_HOME=/opt/gridgain
- ADD in tomcat /usr/share/tomcat6/bin/catalina.sh: CATALINA_OPTS="-DGRIDGAIN_HOME=/opt/gridgain"
- Copy libraries of the WPS framework in the /opt/gridgain/libs/ext/

Other micro instances should be created with the following programs: Java JDK, and GridGain and with the following configuration:

- export GRIDGAIN_HOME=/opt/gridgain
- Copy libraries of the WPS framework in the /opt/gridgain/libs/ext/

[22] The 2433 public instance on February 3, 2011

This instance is used to create a new AMI for launching other instances with the same characteristics. In the section creation of a Grid, other parameters will be added in the configuration of the network.

6.3 Creation of a Grid on the Cloud

The Grid computing paradigm on Cloud has been evaluated using the GridGain extension of 52North. The first task of this extension is to discover the nodes in which the processes can be distributed. This task uses the multicasting[23] network to obtain the available nodes in the network. The AWS platform does not allows working with multicasting over its network. To configure the Grid using Jgroup library[24] the nodes need to be discovered in the AWS. This research works with a Grid controlled by a master node (figure 22) and some nodes that can be added or removed.

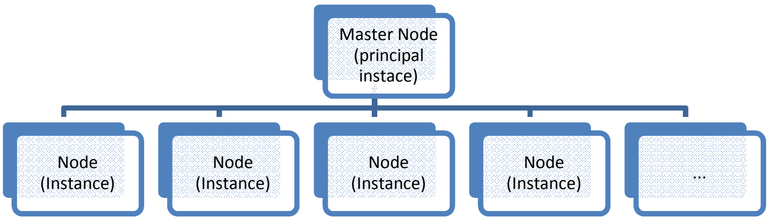


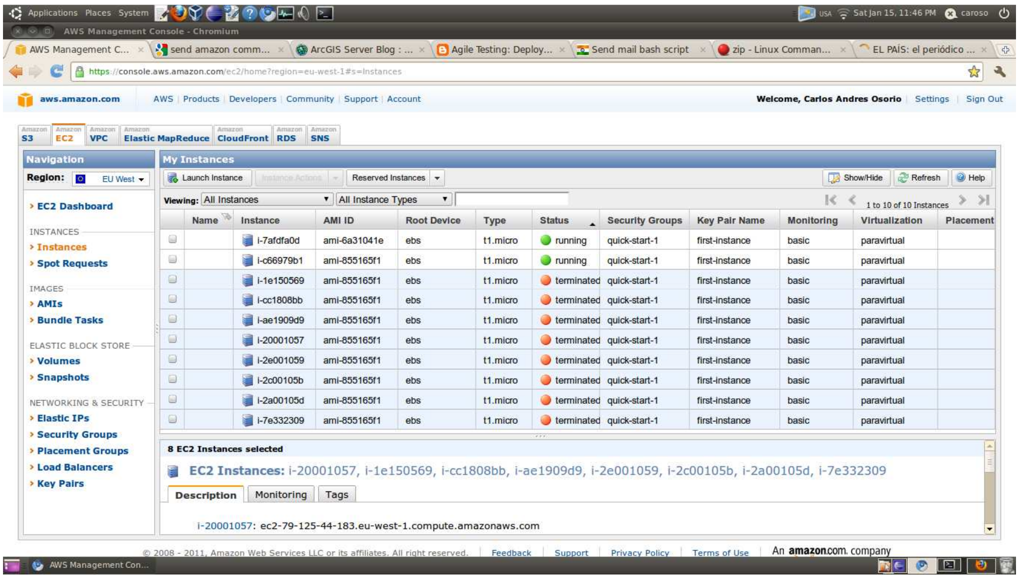
Figure 22. Diagram of nodes used in AWS platform

Another characteristic of AWS is the configuration of the network; each instance in the AWS environment has an external and internal IP. The transmission of data between instances of AWS does not have a cost. The nodes in the Grid are configured using the internal IP. The Master node is configured with a fix IP and the other nodes should include the IP of the master node.

When the normal node has been configured with the Master node IP, it is possible to create an AMI of this node. The configuration of this node will be used by all nodes that are launched with that AMI. In figure 23, the console is shown with 8 stopped

[23] Multicasting: This is used to send simultaneously messages to network of computers
[24] Jgroups: This project is specialized in creation of groups of servers through IP (http://www.jgroups.org)

instances (Node in red). There are two running instances (Nodes in green) that represent the Master node and the base node.



https://console.aws.amazon.com/ec2/home?region=eu-west-1#s=Instances

Welcome, Carlos Andres Osorio

Region: EU West

Viewing: All Instances | All Instance Types

1 to 10 of 10 Instances

	Name	Instance	AMI ID	Root Device	Type	Status	Security Groups	Key Pair Name	Monitoring	Virtualization	Placement
□		i-7afdfa0d	ami-6a31041e	ebs	t1.micro	● running	quick-start-1	first-instance	basic	paravirtual	
□		i-c66979b1	ami-855165f1	ebs	t1.micro	● running	quick-start-1	first-instance	basic	paravirtual	
□		i-1e150569	ami-855165f1	ebs	t1.micro	● terminated	quick-start-1	first-instance	basic	paravirtual	
□		i-cc1808bb	ami-855165f1	ebs	t1.micro	● terminated	quick-start-1	first-instance	basic	paravirtual	
□		i-ae1909d9	ami-855165f1	ebs	t1.micro	● terminated	quick-start-1	first-instance	basic	paravirtual	
□		i-20001057	ami-855165f1	ebs	t1.micro	● terminated	quick-start-1	first-instance	basic	paravirtual	
□		i-2e001059	ami-855165f1	ebs	t1.micro	● terminated	quick-start-1	first-instance	basic	paravirtual	
□		i-2c00105b	ami-855165f1	ebs	t1.micro	● terminated	quick-start-1	first-instance	basic	paravirtual	
□		i-2a00105d	ami-855165f1	ebs	t1.micro	● terminated	quick-start-1	first-instance	basic	paravirtual	
□		i-7e332309	ami-855165f1	ebs	t1.micro	● terminated	quick-start-1	first-instance	basic	paravirtual	

8 EC2 Instances selected

EC2 Instances: i-20001057, i-1e150569, i-cc1808bb, i-ae1909d9, i-2e001059, i-2c00105b, i-2a00105d, i-7e332309

i-20001057: ec2-79-125-44-183.eu-west-1.compute.amazonaws.com

© 2008 - 2011, Amazon Web Services LLC or its affiliates. All right reserved.

Figure 23. Nodes running in the AWS console

6.4 Evaluating of WPS on the Cloud

In this section the framework for the evaluation of the implementation on the Cloud is presented. The topics to evaluate are:

- Relation between the number of nodes and the duration of interpolation process.
- Relation between the amount of data and the number of nodes

The environment of the evaluation:

- Period of evaluation: 2011-02-22 T 9:00:00 Z - 2011-02-22 T 22:00:00 Z
- Dataset used size: Data elevation 10000 points.
- The WFS service is located in the same Master Node.
- The spatial resolution requests over the area of interest: 2, 5 and10 meters.
- Number of nodes (micro instances): 1-10
- Number of repetitions: 10
- The evaluation is executed sequentially with intervals of 15 seconds.
- The requests are generated randomly.

- Number of evaluations per method: 300 (10 nodes x 3 resolutions x 10 repetitions)

7. Results and discussion

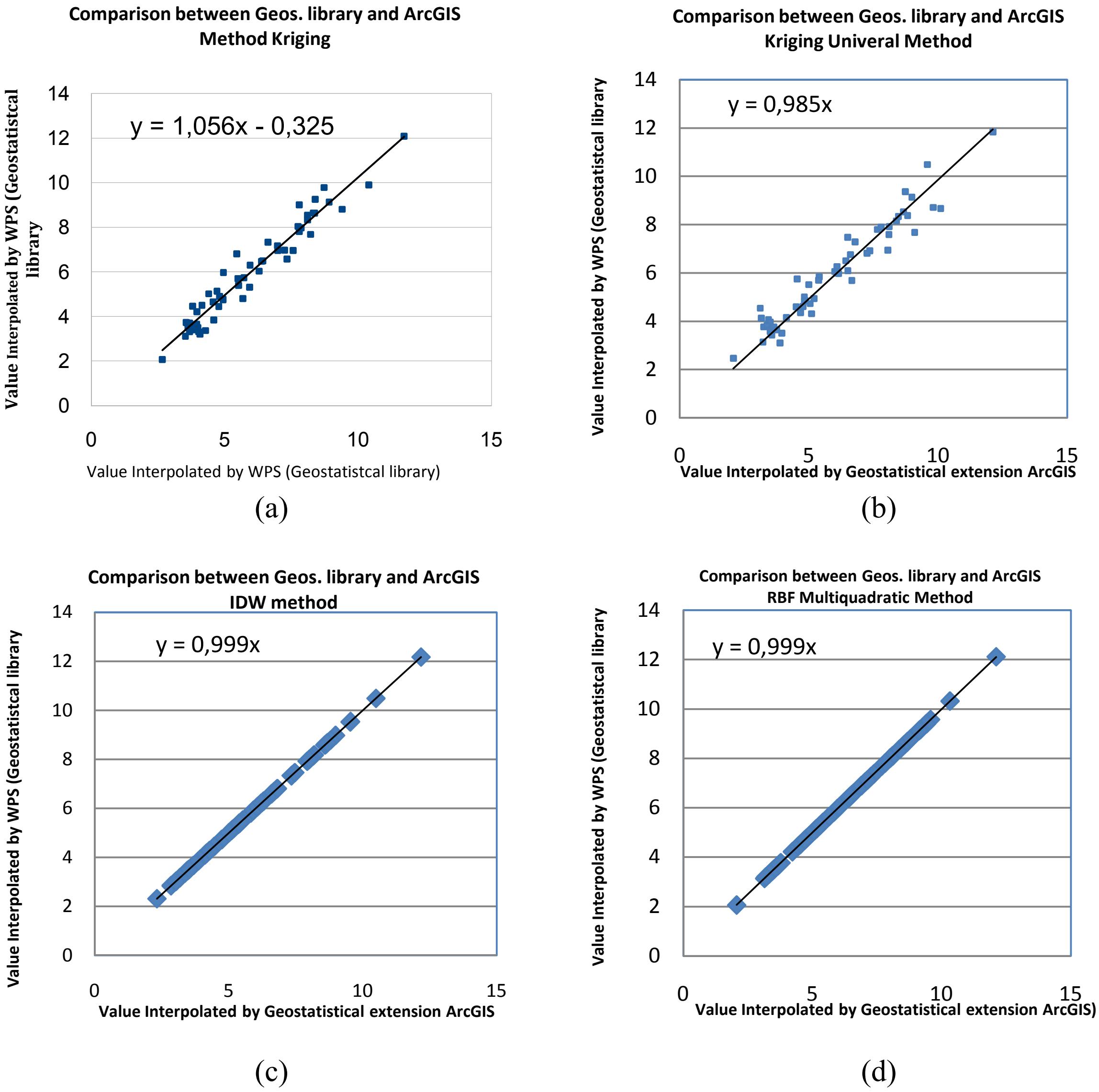
7.1 Evaluation of the Geostatistical Library

The Geostatistical library is evaluated using the maximum daily temperature in Spain dataset described in the previous section and compared with the ArcGIS Geostatistical Extension. The dataset is divided randomly in training (512 stations) and testing (57 stations); with the training dataset the interpolation is executed. The testing dataset is used to validate the exactitude of the interpolation through RMS and standardized RMS. Four interpolation methods are evaluated in each application for determining the differences with the real value (Table 6). The same parameters are used in both applications to interpolate the dataset.

Method	Geostatistical Library		ArcGIS (Geostatistical Extension)	
	RMS	Standardized RMS	RMS	Standardized RMS
IDW *Power=2*	1.607	-	1.607	-
Ordinary Kriging *Exponential(R:261138;S:5.4;N:1.97)*	1.587	0.881	1.605	0.829
Universal Kriging *Exponential(R:29138;S:4.4;N:1.67)*	1.593	0.873	1.623	0.820
RBF *Model: Multiquadratic; Factor=0*	1.648	-	1.648	-

Table 6. Validation of Geostatistical library

The interpolation results of each application are organized in scatter plots by the method of interpolation. The Y-axe describes the values obtained by the Geostatistical library and the X-axe describes the values obtain by ArcGIS (Figure 24).



(a)

(b)

(c)

(d)

Figure 24. Comparison of the Interpolation Methods between ArcGIS and the Geostatistical library

Discussion

The values interpolated with the methods IDW and RBF (Multiquadratic) are quite similar between both applications. The slopes line in both graph (Figure 24c , 24d) are very close to 1. The results confirm that the Geostatistical library executes the interpolation in the same way as the extension Geostatistical Analyst of ArcGIS.

However, the RMS and standardized RMS errors using the Ordinary Kriging method differs between both applications in 0.018 and 0.052 respectively. These differences are associated with internal parameters used by ArcGIS that are not used by the Geostatistical library such as: binding process, selection of neighbor by sectors, etc.

However, the interpolated values in the scatter plot return the slope 1.056 (Figure 24a), confirming that the interpolation with the Geostatistics library is quite similar to the interpolation executed by ArcGIS. The Universal Kriging method describes a difference of 0.03 and 0.053 in the RMS error and standardized RMS respectively between both applications. The scatter plot between the values of interpolation shows a slope of 0.985 (Figure 24b) which it is quite close to 1. The maximum difference between RMS errors of all methods is 0.061 which indicates that the values of interpolation does not vary much, in figure 25 some of the maps with maximum differences are shown.

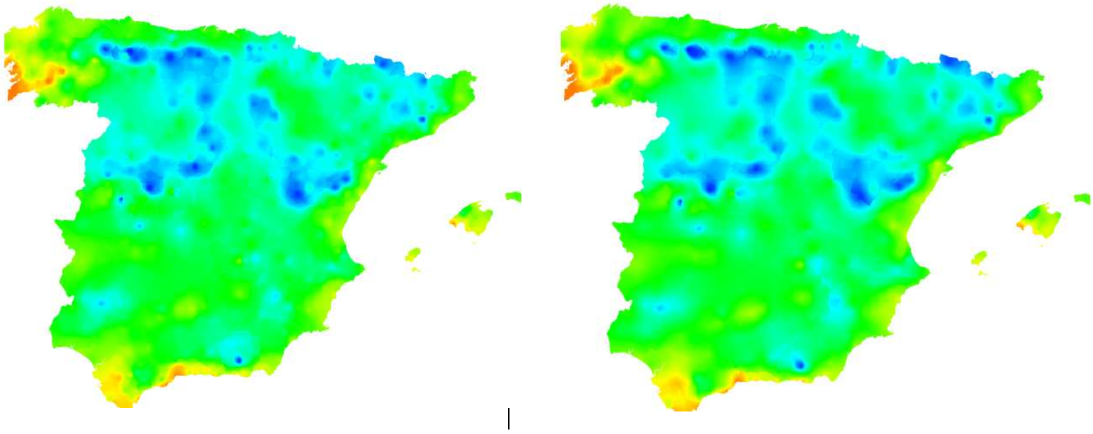
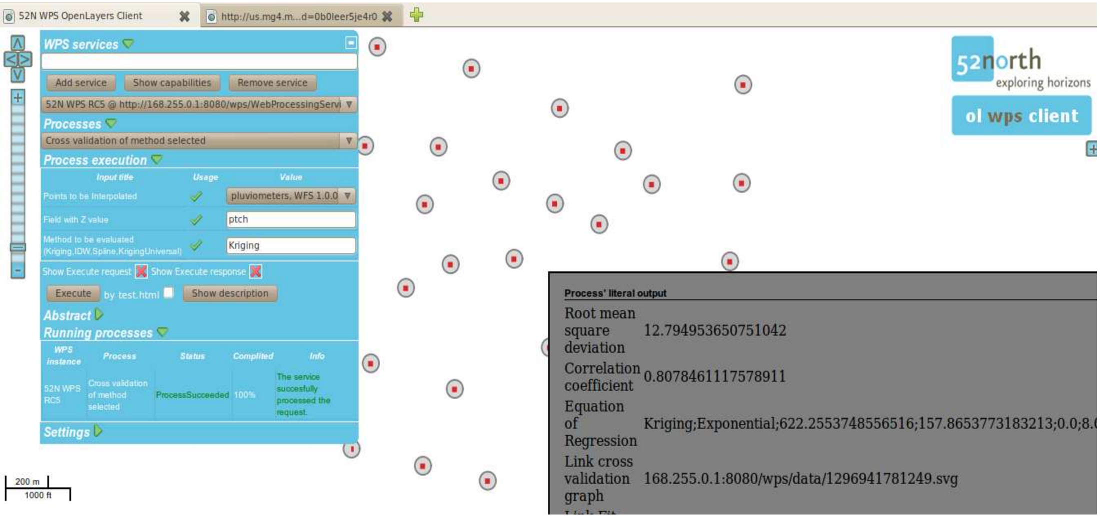


Figure 25. Differences between RBF and Ordinary Kriging

7.1.1 Testing the services on the WPS Client

The WPS services created in this project with Geostatistical library are: Fitting best parameters for an interpolator method and Interpolate. Figure 26 shows the result of a request through 52North WPS Client with the service “Fitting best parameter”. This WPS requires introducing the data, field and the type of interpolation method. Finally, this service returns an array with the Errors, Parameters and graphics related with the method of interpolation selected i.e., Figure 27 are the cross validation and semivariogram generated by the Geostatistical library.



Input title	Usage	Value
Points to be Interpolated	✓	pluviometers, WFS 1.0.0
Field with Z value	✓	ptch
Method to be evaluated (Kriging,IDW,Spline,KrigingUniversal)	✓	Kriging

WPS instance	Process	Status	Complited	Info
52N WPS RC5	Cross validation of method selected	ProcessSucceeded	100%	The service succesfully processed the request.

Process' literal output

Root mean square deviation	12.794953650751042
Correlation coefficient	0.8078461117578911
Equation of Regression	Kriging;Exponential;622.2553748556516;157.8653773183213;0.0;8.(
Link cross validation graph	168.255.0.1:8080/wps/data/1296941781249.svg

Figure 26. Finding the best parameters

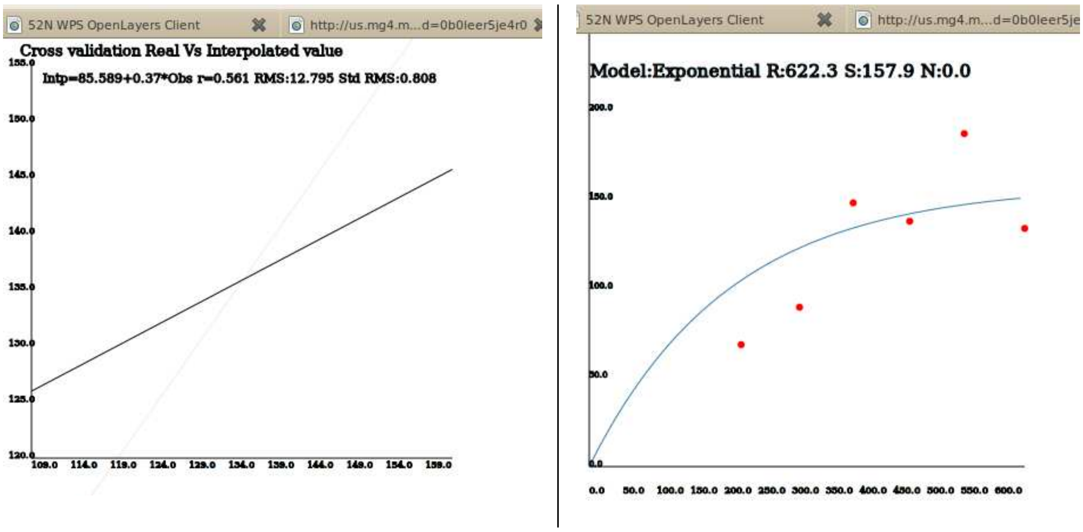
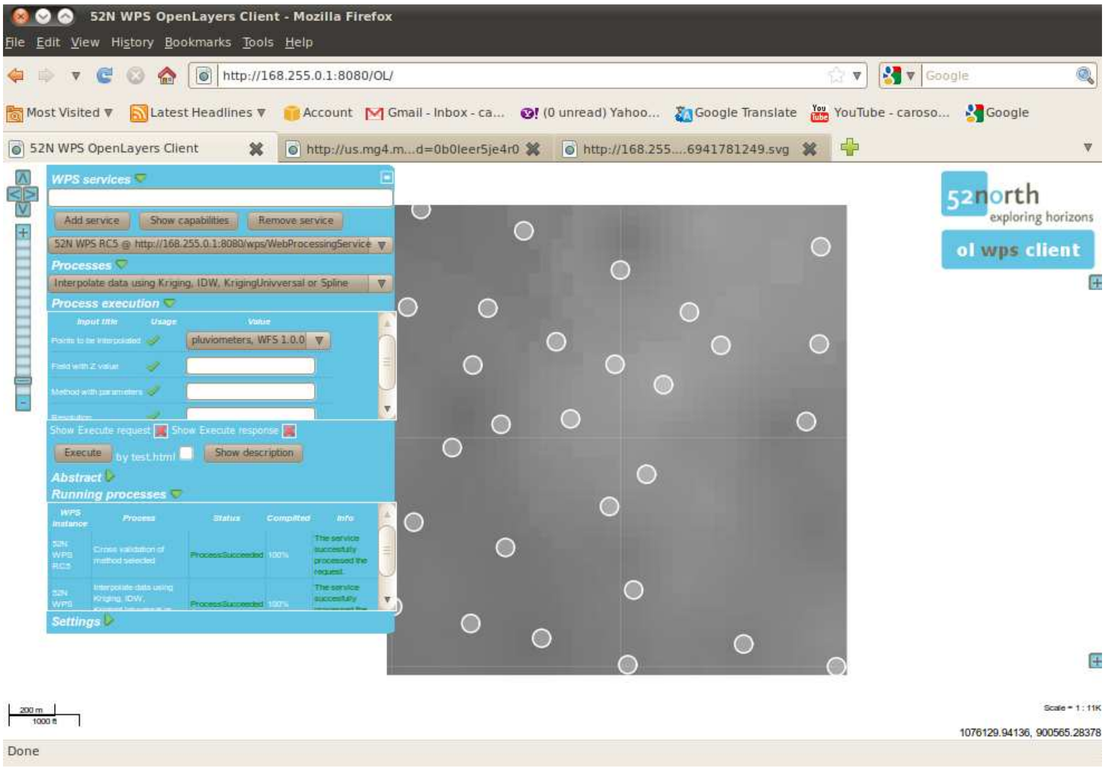


Figure 27. Graphics of cross validation and semivariogram generated by the Geostatistical library

The second implemented service executed the interpolation process (figure 28); the WPS service requires introducing the data, field, Interpolation method with its parameters and the resolution of the raster generated. The result of the service is a GeoTiff image which it is stored in Geoserver. The WPS service returns a reference to this GeoTiff through a WMS published in Geoserver. The WMS is used by the 52North WPS client to display the result



http://168.255.0.1:8080/OL/

52N WPS RC5 @ http://168.255.0.1:8080/wps/WebProcessingService

Interpolate data using Kriging, IDW, KrigingUnivversal or Spline

Input title	Usage	Value
Points to be interpolated	✓	pluviometers, WFS 1.0.0
Field with Z value	✓	
Method with parameters	✓	
Resolution	✓	

WPS instance	Process	Status	Completed	Info
52N WPS RC5	Cross validation of method selected	ProcessSucceeded	100%	The service succesfully processed the request
52N WPS	Interpolate data using Kriging, IDW,	ProcessSucceeded	100%	The service succesfully

1076129.94136, 900565.28378

Figure 28. Interpolation executed by the WPS with the Geostatistical library

Discussion

The WPS client is based on an Internet interface allows for loading WPS services satisfactorily. Although, this WPS client can support multiple WPS servers, they should be configured previously. This limitation is related with the restrictions of the JavaScript language to request external information. However, the interface of this client requires loading the data e.g., WFS service, previously to manage the WPS capabilities. The performance of this client is related with the quantity of data that should be loaded in the browser. The WPS specification allows for managing the references in the responses. The interpolation services uses a WFS service as input and a WCS output, but the WPS client just supports WFS responses. It was added the capability to support WMS responses. In this way, the WPS client can visualize the responses of the interpolation service through a reference of a WMS service.

The management of huge datasets in this WPS client would require combining the capabilities of the WMS and WFS to obtain and visualize data through references. The WPS services that require WFS services could accept WMS services with reference to the WFS services. Using this configuration, the WPS client could manage huge datasets.

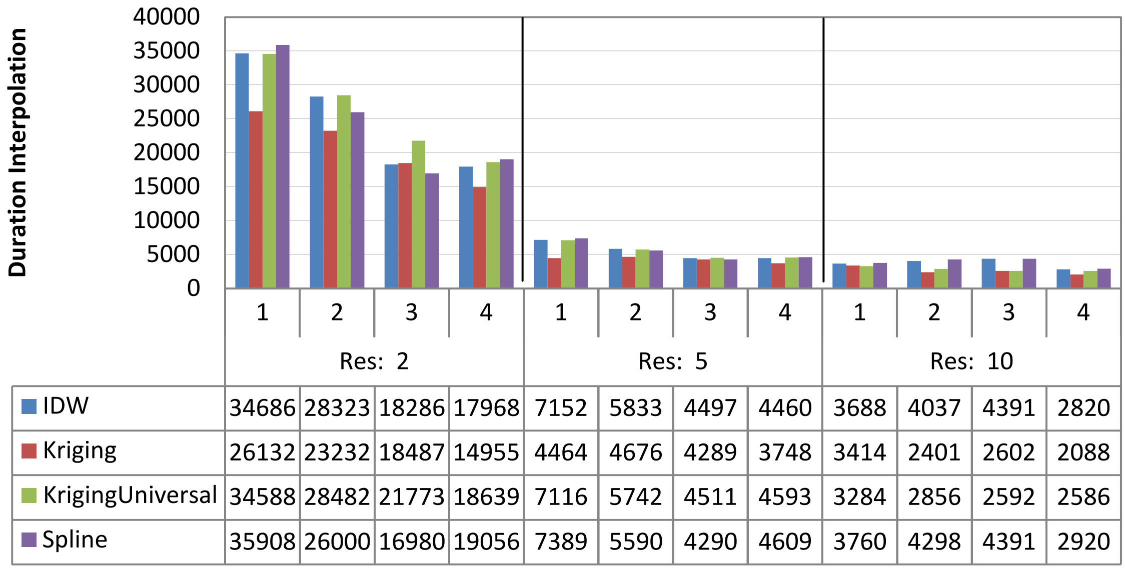
7.1.2 Evaluation of parallelization of WPS in an intranet

The WPS services with Geostatistical capabilities are tested in an internal network to determine the performance of the service. This evaluation is done using the elevation dataset with 1000 points described in the chapter three. Computers with 2 cores are connected through ad hoc to simulate a Grid with four nodes; each core is assigned to one GridGain node. In addition, a WFS service in Geoserver is created for providing the data in SHP-ZIP format.

The WPS in parallel is tested by a Java Client which is configured to send the requests sequentially. The four methods are evaluated with resolution 2, 5 and 10 meters, and incrementing the nodes one by one until four. The interest area is a square of 2 km x 2 km, which indicate that the amount of data to be processed is 1'000.000, 160.000 and 40.000 pixels per resolution respectively. The duration of the process is captured by the Java client; the duration of the process is included in the output of WPS service. In this way, it is possible determining if there are latency problem in the network. The table 6 shows the statistics of the differences between duration requests and processing. The result of the WPS execution in the Grid is showed in the figure 29.

Differences (milliseconds)	
Mean	1604
Median	1445
Mode	1234
Standard Deviation	347
Minimum	1163
Maximum	2364
Suma	76981
Count	48

Table 7. Statistics of the differences between duration requests and processing



	Res: 2				Res: 5				Res: 10			
	1	2	3	4	1	2	3	4	1	2	3	4
🟦 IDW	34686	28323	18286	17968	7152	5833	4497	4460	3688	4037	4391	2820
🟥 Kriging	26132	23232	18487	14955	4464	4676	4289	3748	3414	2401	2602	2088
🟩 KrigingUniversal	34588	28482	21773	18639	7116	5742	4511	4593	3284	2856	2592	2586
🟪 Spline	35908	26000	16980	19056	7389	5590	4290	4609	3760	4298	4391	2920

Figure 29. Distribution duration of interpolation in Grid per number of nodes and resolution spatial

Discussion

According to the evaluation in the table 7, the average of the difference between the requests and processing indicates that the network affects the total duration of all interpolations in 1600 milliseconds approximately. That duration of the process, when the area of interest is worked with a resolution of 2 meters, it is not affected by the network delay. The duration of the interpolation decreases when the number of nodes increases; this behavior is also found by Kerry & Hawick (1997); Pesquer-Mayos (2008); Strzelczyk & Porzycka (2010) on parallel interpolations. When the resolution used is 5 meters, the duration of the process, between one and two nodes are approximately 2000 milliseconds less, but in the Ordinary Kriging method the time increases in 192 milliseconds. When the amount of data decreases to 160.000 pixels (resolution 5 meters) the execution in parallel does not provide benefits.

7.1.3 Evaluation of parallelization of WPS on Amazon AWS

Using the WPS service **General Cross Validation** (Section 5.1.3) the Grid created on the Cloud with the dataset of elevation is evaluated (section 3.2.2). Increasing the number of nodes one by one, it is requested the calculation of the best parameters for the Ordinary Kriging method (figure 30). The parameters of the interpolation found in all cases are: Ordinary Kriging linear model, range: 1255.53, sill: 11963.1, nugget: 0; number of lags: 11; length lag: 125.9; neighbors used: 5.

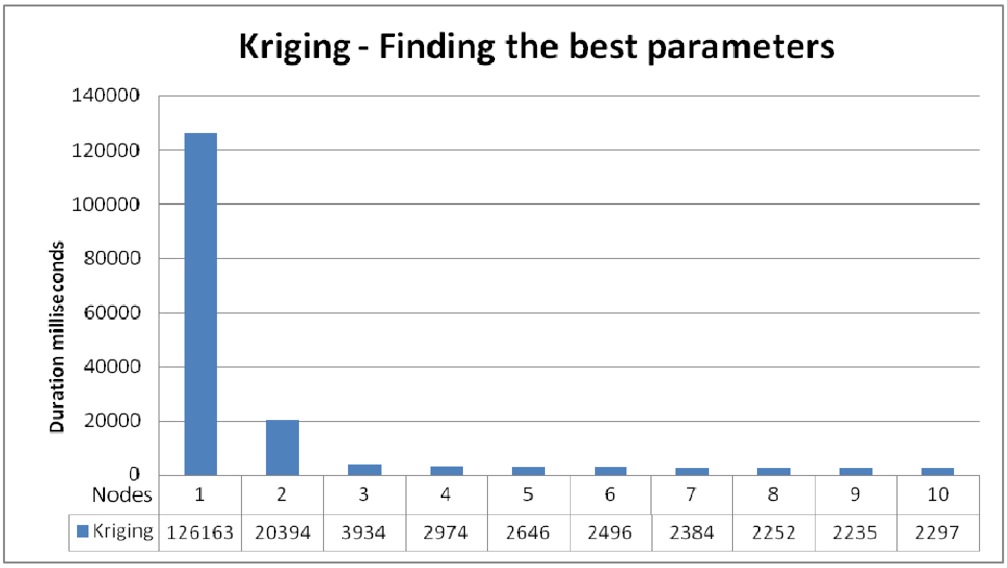


Figure 30. Evaluation WPS general cross validation on the Cloud

Discussion

The dataset (section 3.2.2) is based on a DEM which was interpolated by the Ordinary Kriging method with the theoretical linear model. The WPS service obtained the same model but without nugget effect. The duration of the process decreases when the number of nodes increases. The largest reduction (100 seconds) is observed between one and two nodes. After fourth node is found the threshold in which the number of nodes do not produce an impact over duration of process.

Following the protocol established in the section 6.4, the performance of the parallelization using the WPS service Interpolation (section 5.1.3) on the Cloud with 10.000 points (section 3.2.2) is evaluated. Increasing the number of nodes, one by one, the interpolation with the Ordinary Kriging and IDW methods is requested, using the following parameters: Ordinary Kriging linear model, range: 1255.53, sill: 11963.1, nugget: 0; number of lags: 11; length lag: 125.9; neighbors used: 5, and IDW power 2.4; neighbors used: 5.

Figure 31 shows the variability in the responses time of the evaluations with the Ordinary Kriging and IDW methods , when the numbers of nodes change from 1 to 10 and the spatial resolution takes 2, 5 and 10 meters. When the interpolation time decreases, the number of nodes increases. This behavior is found in all the evaluations.

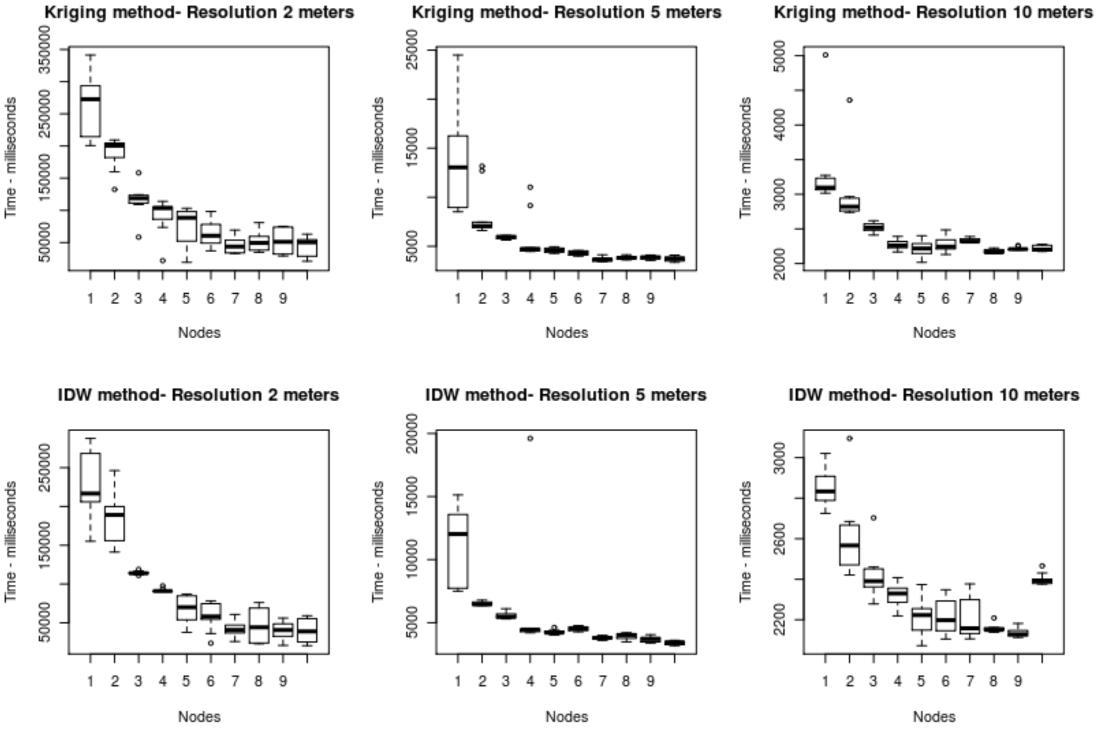


Figure 31. Evaluation parallelization on Amazon AWS

The duration of the interpolation in one node with the Ordinary Kriging and IDW methods in the three resolutions shows high variability, which it is explained by the configuration of the Grid created on the Cloud. The master node used is a micro instance with low capacity. When this master node executes the interpolation process, the duration varies due to memory problems. But, the configuration with more nodes shows that the variability is lower in all cases with both interpolation methods.

Despite of the variability with one node, the time decreases in the interpolation process when more nodes to the process are added. In figure 31, some outlier values which demonstrated memory problems in the master node are showed. The average latency was 640 milliseconds in whole process. This latency is calculated using the difference between processing time and the response time.

In figure 31, the time reduction in the process of interpolation has a limit. Although, the number of nodes increases the time reduction is not appreciable. Using the Analysis of variance and Tukey's Honest Significance Difference test with the software R the time differences are compared. These comparisons are done in each configuration (Ordinary Kriging – Resolution 2m, Ordinary Kriging – Resolution 5m, Ordinary Kriging – Resolution 10m, IDW – Resolution 2m, IDW – Resolution 5m, and IDW – Resolution 10m). The null hypothesis states that duration means from 1 to 10 nodes are the same. The statistical significance between means is evaluated using a t-test, where P<0.001 is considered statistically significant (Table 8). The null hypothesis is rejected in all configurations which indicate that the means are different. This evaluation indicates that there is at least a mean different between the means observed. The Tukey technique allows for determining the statistical significance differences among means.

Nodes	Kriging - 2	Kriging - 5	Kriging - 10	IDW - 2	IDW - 5	IDW - 10
1	265467	14295	3311	227569	11435	2848
2	188513	8205	2981	186288	6512	2605
3	115977	5913	2521	114321	5561	2417
4	92654	5721	2276	91806	5919	2320
5	74576	4594	2207	67028	4249	2218
6	63002	4306	2274	58713	4525	2214
7	46951	3668	2334	41551	3797	2204
8	52448	3840	2175	46951	3915	2157
9	52201	3848	2215	40516	3702	2135
10	43858	3740	2220	39712	3411	2399
F-value	85.766	22.186	23.265	99.45	18.061	56.449
p-value	p<0.001	p<0.001	p<0.001	p<0.001	p<0.001	p<0.001

Table 8 Significance between duration means of each configuration

In table 9 the comparison of the differences in means between nodes of each configuration is showed. The maximum reduction in the duration with the Ordinary Kriging method was 221 seconds, working with ten nodes, and with a resolution of 2 meters. The duration decreased 83% in this configuration. The Ordinary Kriging method with resolution of 5 meters obtained a reduction of 74%, working with seven nodes. A reduction of 34% is obtained in the Ordinary Kriging method, working with eight nodes and with a resolution of 10 meters. The reductions had a significance of 95%.

Comparison of the differences in means between nodes (milliseconds)

Node	Node	Kriging Resolution 2 m	Kriging Resolution 5 m	Kriging Resolution 10 m	IDW Resolution 2 m	IDW Resolution 5 m	IDW Resolution 10 m
1	2	**-76955(a)**	**-6091(a)**	-330	**-41282(a)**	**-4923(a)**	**-244(a)**
1	3	**-149491(a)**	**-8382(a)**	**-791(a)**	**-113249(a)**	**-5874(a)**	**-431(a)**
1	4	**-172814(a)**	**-8575(a)**	**-1035(a)**	**-135764(a)**	**-5516(a)**	**-528(a)**
1	5	**-190891(a)**	**-9702(a)**	**-1104(a)**	**-160542(a)**	**-7186(a)**	**-631(a)**
1	6	**-202466(a)**	**-9989(a)**	**-1037(a)**	**-168856(a)**	**-6911(a)**	**-635(a)**
1	7	**-218516(a)**	**-10628(a)**	**-977(a)**	**-186019(a)**	**-7638(a)**	**-644(a)**
1	8	**-213019(a)**	**-10456(a)**	**-1136(a)**	**-180619(a)**	**-7520(a)**	**-691(a)**
1	9	**-213267(a)**	**-10448(a)**	**-1097(a)**	**-187054(a)**	**-7734(a)**	**-713(a)**
1	10	**-221610(a)**	**-10556(a)**	**-1092(a)**	**-187858(a)**	**-8024(a)**	**-450(a)**
2	3	**-72537(a)**	-2292	**-461(a)**	**-71967(a)**	-951	**-188(a)**
2	4	**-95859(a)**	-2484	**-706(a)**	**-94483(a)**	-593	**-285(a)**
2	5	**-113937(a)**	**-3611(a)**	**-774(a)**	**-119260(a)**	-2263	**-387(a)**
2	6	**-125511(a)**	**-3899(a)**	**-707(a)**	**-127575(a)**	-1988	**-391(a)**
2	7	**-141562(a)**	**-4538(a)**	**-648(a)**	**-144737(a)**	**-2715(a)**	**-401(a)**
2	8	**-136065(a)**	**-4365(a)**	**-806(a)**	**-139338(a)**	**-2597(a)**	**-448(a)**
2	9	**-136313(a)**	**-4357(a)**	**-767(a)**	**-145773(a)**	**-2811(a)**	**-470(a)**
2	10	**-144655(a)**	**-4466(a)**	**-762(a)**	**-146576(a)**	**-3102(a)**	**-206(a)**
3	4	-23323	-193	-245	-22516	357	-97
3	5	**-41401(a)**	-1320	-314	**-47293(a)**	-1313	**-200(a)**
3	6	**-52975(a)**	-1607	-247	**-55608(a)**	-1037	**-204(a)**
3	7	**-69026(a)**	-2246	-187	**-72770(a)**	-1765	**-213(a)**
3	8	**-63529(a)**	-2074	-346	**-67371(a)**	-1647	**-260(a)**
3	9	**-63776(a)**	-2066	-307	**-73806(a)**	-1860	**-282(a)**
3	10	**-72119(a)**	-2174	-302	**-74609(a)**	-2151	-19
4	5	-18078	-1127	-69	-24778	-1670	-103
4	6	-29653	-1415	-2	**-33093(a)**	-1395	-107
4	7	**-45703(a)**	-2054	57	**-50255(a)**	-2122	-117
4	8	**-40206(a)**	-1881	-101	**-44855(a)**	-2005	**-164(a)**
4	9	**-40454(a)**	-1873	-62	**-51290(a)**	-2218	**-186(a)**
4	10	**-48797(a)**	-1982	-57	**-52094(a)**	-2509	78
5	6	-11575	-288	67	-8315	275	-4
5	7	-27625	-927	126	-25477	-452	-14
5	8	-22128	-755	-32	-20078	-335	-61
5	9	-22376	-747	7	-26513	-548	-83
5	10	-30719	-855	12	-27316	-839	**181(a)**
6	7	-16051	-639	59	-17163	-728	-10
6	8	-10554	-467	-99	-11763	-610	-57
6	9	-10802	-459	-60	-18198	-823	-79
6	10	-19144	-567	-55	-19002	-1114	**184(a)**
7	8	5497	172	-159	5399	117	-47
7	9	5249	180	-120	-1036	-96	-69
7	10	-3094	72	-115	-1839	-387	**194(a)**
8	9	-248	8	39	-6435	-214	-22
8	10	-8591	-101	44	-7239	-505	**241(a)**
9	10	-8343	-109	4	-804	-291	**263(a)**
Max difference		**-221610**	**-10628**	**-1136**	**-187858**	**-8024**	**-713**

Table 9 Comparison of the differences in means between nodes. (a) indicates significance > 95%

The maximum reduction in the duration with the IDW method was 188 seconds, with the resolution of 2 meters working with ten nodes. The duration decreased 83% in this configuration. The IDW method with resolution of 5 meters obtained a reduction of 8 seconds (70%), and working with ten nodes. With resolution of 10 meters a reduction of 25%, with nine nodes, is obtained in the IDW method. The reductions had a significance of 95%.

The maximum reduction of time is not related with the optimal number of nodes used to execute the interpolation. In the evaluation of the Ordinary Kriging method with the resolution of 2 meters, after the fifth node, the process did not show any statistical difference. Working with five nodes, the reduction in the duration of the interpolation was 190 seconds (72%) with a statistical significance of the 95%. In the Ordinary Kriging method with resolution of 5 meters, the duration decreased 8 seconds (59%) in the third node. After this node the differences were not significant. The Ordinary Kriging method with resolution of 10 meters obtained a reduction of 0.8 seconds (23%) in the third node. Three nodes was the optimal amount to work with. The optimal number of nodes with the IDW method and resolutions 2, 5, and 10 were 5, 3 and 4 with reductions in the time of 160 seconds (71%), 6 seconds (51%), and 0.5 seconds (19%) respectively.

The amount of pixels that should be processed determines the feasibility of applying the parallelization profile of the WPS service. The interpolation with spatial resolution of 10 meters (40.000 pixels) just reduces the duration of the process in 0.8 seconds in the best case. When the amount of data increases, the parallelization provides better benefits. With the spatial resolution of 5 meters is found a reduction of 8 seconds. The maximum effect in the reduction of time is found with the spatial resolution of 2 meters (1.000.000 pixels) with 221 seconds.

7.1.4 Experiment on the Cloud

According to the previous evaluation, the master node with low capacity produced high variability in the duration of the interpolation. In this way, other experiment with a master node with better capacity (Medium instance) is executed. The experiment plot is:

- Comparison between: One node (High CPU Medium instance) versus 3 nodes, one master node (High CPU Medium instance) and two nodes (micro-instances).
- Dataset used: The resolution is 4 meters (250.000 pixels).
- Repetitions: The interpolation is executed 30 times with 1 node (Master node) and 3 nodes (Master node and two nodes)
- The experiment is executed on January 20 / 2011 between 16:00 – 19:00 UMT.
- The null hypothesis l: The means of the interpolations with the Ordinary Kriging and IDW methods with one node and three nodes are equals

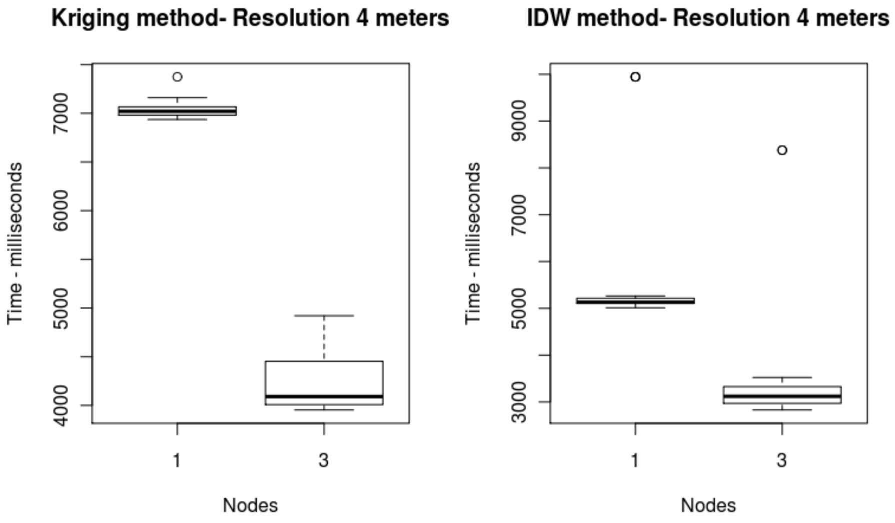


Figure 32 Duration of the interpolation with one and three nodes on the Cloud

	IDW-1	IDW-3	Kriging-1	Kriging-3
Mean	5620	3545	7040	4253
Std dev	1467	1435	85	306
Median	5130	3131	7021	4137
Min	5011	2828	6936	3953
Q1	5107	3037	6983	4008
Q3	5205	3334	7066	4467
Max	9945	8379	7374	4921

Table 10. Statistics interpolation with a master node (Medium instance) and two nodes

Analysis of variance interpolation with IDW method with one node and three nodes:

Grupos	*Count*	*Sum*	*Mean*	*Variance*
IDW-1	30	168603	5620,1	2153472,093
IDW-3	30	103982	3466,066667	1818742,478

ANOVA

Origin	*Sum Sq*	*Df*	*Mean Sq*	*F*	*P-value*	*Critical F*
Between Group	69597894,02	1	69597894,02	35,04236378	1,8484E-07	4,006872822
Intra Group	115194222,6	58	1986107,286			
Total	184792116,6	59				

Analysis of Variance interpolation with Ordinary Kriging method with one node and three nodes:

Grupos	*Count*	*Sum*	*Mean*	*Variance*
Kriging-1	30	211191	7039,7	7302,217241
Kriging-3	29	122987	4240,931034	89013,13793

ANOVA

Origin	*Sum Sq*	*Df*	*Mean Sq*	*F*	*P-value*	*Critical F*
Between Group	115505147,8	1	115505147,8	2434,715845	1,85856E-48	4,009867854
Intra Group	2704132,162	57	47440,91512			
Total	118209279,9	58				

Discussion

The high variability in the duration of the interpolation process with one node is not observed in this experiment. This demonstrates that low capacity in the master node provokes high variability in the process. Also, this experiment has demonstrated that the addition of two interpolation nodes improves the performance of the process. The null hypothesis establishes that there are not any differences in the duration between one node and three nodes with the IDW and Ordinary Kriging methods. The ANOVA analysis showed that the null hypothesis should be rejected because the p-value <0.05 in both cases. The interpolation is executed 36% faster in the case of IDW when two nodes are added to the master node and 40% faster in the Ordinary Kriging. That experiment proves that the grid configuration on the Cloud can increase the capabilities of geoprocessing.

7.2 Cost evaluation

This thesis worked with a free account of Amazon AWS that provides a micro instance for one year free. The quantity of information used by this thesis never was higher than the limits established for a free account. The micro instances in Amazon can run during 730 hour per month free. Amazon sums the quantity of hours that a micro-instance has run.

The service EBS allows storing 10 GB without extra cost. That capacity was enough for this research work. In this thesis was used a small instance ($0.095/ hour) to test its performance during 24 hours and a Medium instance during 5 hours($0.38/hour).

The nodes used in the Grid evaluation did not generate any cost because the number of hours running did not exceed the hours allowed by the account. Also, the communications between instances of the Amazon AWS do not generate any cost. The bill associated was $4.20 ~ €3.09 Appendix (E)

8. Conclusion and future work

The geoprocessing on parallel have been used since 1960's with the invention of computer for solving complex problems and operations, but these applications and infrastructures were designed only for scientific projects. With the expansion of the Cloud Computing technologies, the applications and new software can execute complex tasks in Internet without limitation of resources and at a reasonably cheap cost. The geoprocessing can use the infrastructure of Cloud Computing to provide services with similar capabilities than stand alone software. This thesis contributes in the generation of alternatives for processing geospatial data in the special case of Geostatistics. The library implemented four interpolation methods (Ordinary Kriging, Universal Kriging, IDW and RBF) with a model to find the parameter for each method. This application does not have any differences in the methods IDW and RBF with control software (ArcGIS Gestatistical Analyst extension). Otherwise, the Ordinary and Universal Kriging methods had differences of 0.018 and 0.03 in the

RMS error respectively with the error obtained by the control software. These differences should be studied in a future work.

The OGC WPS interface facilitated the implementation of the library on Internet through the 52North WPS framework. This framework allowed adding the parallelization features to the Geostatistical library through its extension GridGain. The WPS services generated with the framework were evaluated on the Amazon AWS to test the performance of the parallelization profile on the Cloud. The 52North WPS OpenLayer client was used to execute the geoprocesses. This WPS client was configured to accept responses with references to WMS services. This WPS client required loading the data in the browser which reduced its performance with huge datasets. In the evaluation of the performance of the Cloud, a Java client was used.

The evaluation of the parallelization of the WPS services was tested in an Intranet with four nodes. The parameters used for evaluating the performance were the number of nodes and the amount of data to be processed. The variable measured was the duration of the process. The results demonstrated that duration of the interpolation process decreases when the number of nodes increases, the reduction of the time with four nodes was approximately 48% and 43% in the IDW and Ordinary Kriging method respectively.

Using 10 micro instances in the Amazon AWS, the duration of the process on the Cloud was evaluated. The parallelized WPS with the 52North framework allowed for executing the interpolations following the same procedure done in the Intranet. The WPS service that determines the best parameters for each interpolation method was parallelized. The major reduction in the duration of this WPS service was found with three nodes. Besides, the interpolation process on the Cloud was evaluated with several configurations. The configuration with 1.000.000 pixels (resolution 2 meters) and a dataset of 10.000 points was found as the higher reduction in the duration of the interpolation process. The duration of the process decreased in 72% and 71% with the Ordinary Kriging and IDW methods respectively with a statistical significance of 95%, and in both cases the optimal number of nodes was five. The interpolation process on the Cloud also showed a high variability in the duration with one node due to the low capacity of the master node used. To demonstrate that the variability was caused by the master node, other experiment was planned, where a low variability in the duration of the process, with a master node with high capacity, was found.

Finally, the experiments demonstrated that the geoprocessing on the Cloud of GI is feasible through the WPS interface. The performance of some WPS services, with geostatistical methods especially, can be improved by the parallelization technique. This thesis shows that the parallelization on the Cloud is viable using a Grid configuration. Furthermore, the evaluation showed that the parallelization of geoprocesses on the Cloud for academic purposes is inexpensive using the Amazon AWS platform.

.

Future work

The future work of this thesis can be divided into two topics: the Geostatistics library, and WPS on Cloud. There are some aspects that can be added to the library such as capabilities to support other complex interpolation methods, e.g., co-Kriging, fractals, etc. The analysis of the differences with other control software should be conducted to improve the Geostatistical library. Finally, the option of adding variable selections should be used for neighbors and anisotropy analysis in the Kriging methods.

The geoprocessing on the Cloud can use the capabilities of scaling and load balancing of the Cloud to provide a better quality of service. The storage of Geographic information on the Cloud is a fundamental topic that should be developed to increase the capabilities of geoprocessing.

References

Alonso-Calvo, R., Crespo, J., García-Remesal, M., Anguita, A., & Maojo, V. (2010). On distributing load in Cloud Computing: a real application for very-large image datasets. *Procedia Computer Science,* May, 1 (1), 2669–2677. doi:10.1016/j.procs.2010.04.300

Amazon. (2010). AWS. Public Data Sets on AWS. Available from: http://aws.amazon.com/publicdatasets/.

Armbrust, M., Fox, A., Griffith, R., Joseph, A.D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I., Zaharia, M. (2009). Above the Clouds: a Berkeley view of Cloud Computing. *Technical Report. EECS Department*, University of California, Berkeley. Retrieved from http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.pdf

Aymerichi, F.M., Fenu, G., & Surcis, S. (2008). An approach to a Cloud Computing network. *First International Conference on the Applications of Digital Information and Web Technologies (ICADIWT 2008)*, August 4-6, Ostrava, Czech Republic, 113-118. doi: 10.1109/ICADIWT.2008.4664329

Baranski, B. (2008). Grid Computing enabled Web Processing Service. In: E. Pebesma, M. Bishr & T. Bartoschek (Eds.), *GI-Days 2008 - Proceedings of the 6th Geographic Information Days*. IfGIprints, 32, Institute for Geoinformatics, University of Münster

Blower, J.D. (2010). GIS in the Cloud: implementing a web map service on Google App Engine. *Proceedings of the 1st International Conference and Exhibition on Computing for Geospatial Research & Application (COM.Geo'10)*, June 21-23, Washington, DC. doi: 10.1145/1823854.1823893

Boss, G., Malladi, P., Quan, D., Legregni, L., & Hall, H. (2007). Cloud Computing. *IBM Corporation. High Performance on Demand Solutions*, October. Retrieved from http://download.boulder.ibm.com/ibmdl/pub/software/dw/wes/hipods/Cloud_computing_wp_final_8Oct.pdf

Brauner, J., Foerster, T., Schaeffer, B., & Baranski, B. (2009). Towards a research agenda for geoprocessing services. *12th AGILE International Conference on Geographic Information Science*, Leibniz Universität Hannover, Germany. Retrieved from http://www.ikg.uni-hannover.de/agile/fileadmin/agile/paper/124.pdf

Cary, A., Yesha, Y., Adjouadi, M., & Rishe, N. (2010). Leveraging Cloud Computing in geodatabase management. *IEEE International Conference on Granular Computing (GrC),* August 14-16, San Jose, California, 73-78. doi: 10.1109/GrC.2010.163

Chang, K.T. (2004). *Introduction to Geographic Information Systems*. (2nd ed.). New York, NY: McGraw-Hill, Inc.

Chappell, D. (2009). *Introducing the windows azure platform*. San Francisco, California: Chappell & Associates. White Papers. Retrieved from http://www.davidchappell.com/writing/white_papers/Windows_Azure_Platform_v1.3--Chappell.pdf

Cui, D., Wu, Y., & Zhang Q. (2010). Massive spatial data processing model based on Cloud Computing model. *Third International Joint Conference on Computational Science and Optimization (CSO),* May 28-31, Huangshan, Anhui, 2, 347-350. doi: 10.1109/CSO.2010.168

Dean, J., & Ghemawat S. (2004). MapReduce: simplified data processing on large cluster. *Proceedings of 6th Symposium on Operating Systems Design and Implementation (OSDI'04),* December 6-8, San Francisco, California, 137-149. Retrieved from http://www.usenix.org/events/osdi04/tech/full_papers/dean/dean.pdf

Díaz, L., Granell, C., & Gould, M., (2008). Case study: geospatial processing services for web-based hydrological applications. In J.T. Sample, K. Shaw, S. Tu, M. Abdelguerfi (Eds.), *Geospatial Services and Applications for the Internet.* Springer. Retrieved from http://www.geoinfo.uji.es/pubs/2008-BookChapter_submitted.pdf

Fitch, P., & Bai, Q. (2009). A standards based web service interface for hydrological models. *18th World IMACS / MODSIM Congress*, July 13-17, Cairns, Australia.

Foster, I., Zhao, Y., Raicu, I., & Lu, S. (2008). Cloud Computing and Grid Computing 360-degree compared. *Grid Computing Environments Workshop (GCE'08)*, November 12-16, Austin, Texas. Retrieved from: http://people.cs.uchicago.edu/~yongzh/pub/GCE08_Cloud_Grid.pdf

Gong, C., Liu, J., Zhang, Q., Chen, H., & Gong, Z. (2010). The characteristics of Cloud Computing. *39th International Conference on Parallel Processing Workshops (ICPPW'10)*, September 13-16, San Diego, California, 275-279. doi: 10.1109/ICPPW.2010.45

GridGain. (2010). *Cloud application platform.* Retrieved from http://www.gridgain.com

Grossman, R.L. (2009). The case for Cloud Computing. *IT Professional,* March/April, 11(2), 23-27. doi: 10.1109/MITP.2009.40

Hadoop. (2009). *MapReduce tutorial*. Retrieved from http://hadoop.apache.org/mapreduce/docs/current/mapred_tutorial.pdf

Hawick, K.A., Coddington, P.D., & James, H.A. (2003). Distributed frameworks and parallel algorithms for processing large-scale geographic data. *Parallel Computing (Special Issue on High Performance Computing with Geographic Data)*, October, 29(10), 1297-1333. doi:10.1016/j.parco.2003.04.001

Jinnan, Y., & Sheng, W. (2010). Studies on application of Cloud Computing techniques in GIS. *Second IITA International Conference on Geoscience and Remote Sensing,* August 28-31, Quingdao, China, 1, 492-495. doi: 10.1109/IITA-GRS.2010.5602628

Johnston, K., ver Hoef, J.M., Krivoruchko, K., & Lucas, N. (2001). *Using ArcGis geostatistical analyst.* Redlands, CA: Esri Press.

Karayusufoglu, S., Eris, E., & Coskun, G. (2010). Estimation of basin parameters and precipitation distribution of Solakli Basin, Turkey. *WSEAS Transactions on Environment and Development*, May, 6 (5), 385-394. Retrieved from http://www.wseas.us/e-library/transactions/environment/2010/89-589.pdf

Keahey, K., Figueiredo, R., Fortes, J., Freeman, T., & Tsugawa, M. (2008). Science Clouds: early experiences in Cloud Computing for scientific applications. *Cloud Computing and its Applications Workshop (CCA'08)*, October 22-23, Chicago, Illinois. Retrieved from http://www.cca08.org/papers/Paper39-Kate-Keahey.pdf

Kerry, K.E., & Hawick, K.A. (1997). *Spatial interpolation on distributed, high-performance computers* (DHPC Technical Report DHPC-015). Adelaide, Australia: Department of Computer Science, University of Adelaide.

Ku, W-Y., Chou, T-Y., & Chung L-K. (2010). Hadoop on architecture design of GIS Cloud Computing. *Asia GIS 2010 International Conference GIS & Cloud Computing*. November 5-6, Kaohsiung, Taiwan. Retrieved from: http://www.agis2010.tgic.org.tw/fulltext/Nov.4/A/A3/4R302A03.pdf

Ladra, S., Rodríguez-Luaces, M., Pedreira, O., & Seco D. (2008). A toponym resolution service following the OGC WPS standard. *The 8th international Symposium on Web and Wireless Geographical Information Systems (W2GIS`08)*, December 11-12, Shanghai, China, 75-85.

lat/lon GmBH. (2009). deegree Web Processing Service v2.4. Retrieved from http://download.deegree.org/deegree2.4/docs/wps/deegree_wps_documentation_en.pdf

Liu, J., & Liu, P. (2010). Status and key techniques in Cloud Computing. *3rd International Conference on Advanced Computer Theory and Engineering (ICACTE)*, August 20-22, Chengdu, China, 4, 2154-7491. doi: 10.1109/ICACTE.2010.5579728

Lu, G.Y., & Wong, D.W. (2008). An adaptative inverse-distance weighting spatial interpolation technique. *Computers and Geosciences*, 34, 1044-1055. doi: 10.1016/j.cageo.2007.07.010

Mahanti, A., & Eager, D. (2004). Adaptive data parallel computing on workstation clusters. *Journal of Parallel and Distributed Computing*, November, 64(11), 1241-1255.

Mikkilineni, R., & Sarathy, V. (2009). Cloud Computing and the lessons from the past. *18th IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises (WETICE'09)*, June 29-July 1, Groningen, The Netherlands, 57-62. doi: 10.1109/WETICE.2009.14

Napper, J., & Bientinesi. (2009). Can Cloud Computing reach the TOP500? *Second Workshop on Unconventional High-Performance Computing (UCHPC'09),* May. doi: 10.1145/1531666.1531671

Nash, E., Bobert, J., Wenkel, K-O., Mirschel, W., & Wieland, R. (2007). Geocomputing made simple: service-chain based automated geoprocessing for precision agriculture. *Proceedings of GeoComputation*, Maynooth, Ireland. U. Demšar (Ed.). National University of Ireland, Maynooth.

Open Geospatial Consortium. (2005). *OWS-4 Geo Processing Workflow (GPW)*. Open Geospatial Consortium Inc. Retrieved from http://www.ogcnetwork.net/node/233

Open Geospatial Consortium. (2007). *OpenGIS Web Processing Service. OGC implementation specification.* Open Geospatial Consortium Inc. Retrieved from http://www.opengeospatial.org/standards/wps

Open Geospatial Consortium. (2008): *OGC Reference Model*. Open Geospatial Consortium Inc. Reference number: OGC 08-062r4 Version: 2.0. Retrieved from http://www.opengeospatial.org/standards/orm

Pascoe, S., Stephens, A., Alderson, D., Norton, P., James, P., Abele, S., & Iwi, A. (2009). The UK climate projections user interface: a case study for the deployment of a scalable web-application built upon the Open Source Geo-stack and OGC standards. *The UK e-Science All Hands Meeting*, December 7-9, Oxford, UK. Retrieved from http://www.allhands.org.uk/2009/09/PascoeAHM2009UKCP09.pdf

Pautasso, C., & Alonso, G. (2006). Parallel Computing patterns for Grid workflows. *Proceedings of the HPDC2006 Workshop on Workflows in Support of Large-Scale Science (WORKS06),* June 19-23, Paris, France.

Pesquer-Mayos, L. (2008). *Parallelized solution of interpolation Kriging with an automated fitting of the variogram* (master's thesis). Universitat Autònoma de Barcelona, Barcelona, España.

PyWPS development team. *Python Web Processing Service (PyWPS).* Retrieved from http://pywps.wald.intevation.org/index.html

Randles, M., Lamb, D., & Taleb-Bendiab, A. (2010). A comparative study into distributed load balancing algorithms for Cloud Computing. *Proceedings of the 24th International Conference on Advanced Information Networking and Applications (AINA)*, April 20-23, Perth, Australia, 551-556. doi: 10.1109/WAINA.2010.85

Resende, C.M. (2010). *Ambiente Grid utilizando software livre*. (Postgraduate thesis). Universidade Federal de Lavras, Minas Gerais, Brasil. Retrieved from http://www.ginux.ufla.br/files/mono-CristianoResende.pdf

Rimal, B.P., Choi, E., & Lumb, I. (2009). A taxonomy and survey on Cloud Computing systems. *Fith International Joint Conference on INC, IMC and IDC*, 25-27 August, Seoul, Korea, 40-51. doi: 10.1109/NCM.2009.218

Schäffer, B., Baranski, B., & Foerster, T. (2010). Towards Spatial Data Infrastructures in the Clouds. In M. Painho, M. Santos, & H. Pundt (Eds.), *Geospatial Thinking, Lecture Notes in Geoinformation and Cartography* (pp. 399-

418). Held at The 13th AGILE International Conference on Geographic Information Science, Guimarães, Portugal: Springer Verlag.

Sedgewick, R., & Wayne, K. (2010). *Algorithms*. Retrieved from http://algs4.cs.princeton.edu/92search/

Singh-Yadav, S., & Wen-Hua Z. (2010). CLOUD: A Computing infrastructure on demand. *2nd International Conference on Computer Engineering and Technology (ICCET`10)*, April 16-18, Chengdu, China, 1, 423-426. doi: 10.1109/ICCET.2010.5486068

Stollberg, B., & Zipf, A. (2009). Development of a WPS process chaining tool and application in a disaster management use case for urban areas. *27th Urban Data Management Symposium (UDMS 2009),* Ljubljana, Slovenia. Retrieved from http://koenigstuhl.geog.uni-heidelberg.de/publications/bonn/conference/Paper42-UDMS-StollbergZipf.pdf

Stollberg, B., & Zipf, A. (2008). Geoprocessing services for spatial decision support in the domain of housing market analyses - experiences from applying the OGC Web Processing Service interface in practice. *11th AGILE International Conference on Geographic Information Science*, University of Girona, Spain. Retrieved from http://plone.itc.nl/agile_old/Conference/2008-Girona/PDF/103_DOC.pdf

Strzelczyk, J., & Porzycka, S. (2010). Parallel kriging algorithm for unevenly spaced data. *Para 2010 – State of the Art in Scientific and Parallel Computing- extended abstract no. 81*, June 6-9, University of Iceland, Reykjavik. Retrieved from http://vefir.hi.is/para10/extab/para10-paper-81.pdf

52 North. (2010). *Geoprocessing*. Retrieved from http://52north.org/maven/project-sites/wps/52n-wps-webapp/

Talia, D., & Trunfio P. (2010). How distributed data mining tasks can thrive as knowledge services. *COMMUNICATIONS of the ACM,* July, 53 (7), 132-137. doi: 10.1145/1785414.1785451

Tu, S., Flanagin, M., Wu, T., Abdelguerfi, M., Normand, E., & Mahadevan, V. (2004). Design strategies to improve performance of GIS Web Services. *Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC’04)*, April 5-7, Las Vegas, Nevada, 2, 444-448. doi: 10.1109/ITCC.2004.1286692 van Nieuwpoort, R.V. (2003). *Efficient Java-Centric Grid-Computing.* (Doctoral thesis). Vrije Universiteit, Amsterdam, The Netherlands. Retrieved from http://www.cs.vu.nl/~rob/thesis/thesis.pdf

Vaquero, L.M., Rodero-Merino, L., Caceres, J., & Lindner, M. (2009). A break in the Clouds: towards a Cloud definition. *ACM SIGCOMM Computer Communication Review,* 39(1), 50-55. doi: http://doi.acm.org/10.1145/1496091.1496100

Velte, A.T., Velte, T.J., & Elsenpeter, R. (2010*). Cloud Computing- a practical approach.* New York, NY: McGraw-Hill, Inc.

Vouk, M.A. (2008). Cloud Computing– issues, research and implementations. *Proceedings of the 30th International Conference on Information Technology Interfaces (ITI'08)*, June 23-26, Cavtat, Croatia, 31-40. doi: 10.1109/ITI.2008.4588381

Wehrmann, T., Gebhardt, S., Klinger, V., & Künzer, C. (2010). Web services enabled architecture coupling data and functional resources. *Geospatial Data an Geovisualization: Environment, Security and Society: A special joint symposium of ISPRS Technical Comission IV & AutoCarto in conjunction with ASPRS/CaGIS 2010*, November 15-19, Orlando, Florida. Retrieved from http://www.asprs.org/publications/proceedings/orlando2010/files/WEHRMANN.PDF

Wu, J., Ping, L., Ge, X., Wang, Y., & Fu, J. (2010). Cloud storage as the infrastructure of Cloud Computing. *International Conference on Intelligent Computing and Cognitive Informatics (ICICCI 2010)*, June 22-23, Kuala Lumpur, Malaysia, 380-383. doi: 10.1109/ICICCI.2010.119

Xu, D. (2010). Cloud Computing: an emerging technology. *International Conference On Computer Design And Appliations (ICCDA 2010)*, June 25-27, Qinhuangdao, China, 1, 100–104. doi: 10.1109/ICCDA.2010.5541105

Yixin, Z. (2010). A new online trading platform based on Cloud Computing. *Second IITA International Conference on Geoscience and Remote Sensing*, August 28-31, Quingdao, China, 85-88. doi: 10.1109/IITA-GRS.2010.5603271

Yuan, M. (2007). Temporal GIS for agent-based modeling of complex spatial systems. *NSF Research Workshop on Agent-Based Modeling of Complex Spatial Systems,* April 13-15, Santa Barbara, CA. Retrieved from http://www.ncgia.ucsb.edu/projects/abmcss/docs/yuan_paper.pdf

Zhang, S., Chen, X., Zhang, S., & Huo, X. (2010a). The Comparison between Cloud Computing and Grid Computing. *International Conference on Computer Application and System Modeling (ICCASM 2010)*, October 22-24, Taiyuan, China, 11, 72 -75. doi: 10.1109/ICCASM.2010.5623257

Zhang, S., Zhang, S., Chen,, X., & Huo, X. (2010b). Cloud Computing research and development trend. *Second International Conference on Future Networks (ICFN'10)*, January 22-24, Sanya, Hainan, 93-97. doi: 10.1109/ICFN.2010.58

Appendix A

Theoretical models by Johnston et al. (2001)

Nugget effect

The semivariogram model is

$$\gamma(h;\theta)= \begin{cases} 0 \\ \theta_s \ \text{ for } \ \mathrm{h}=0 \end{cases}$$

Where $\theta_s \geq 0$ for $\mathrm{h} \neq 0$

Circular

$$\gamma(h;\theta)=\begin{cases} \dfrac{2\theta_5}{\pi}\left[\dfrac{\|h\|}{\theta_r}\sqrt{1-\left(\dfrac{\|h\|}{\theta_r}\right)^2}+\arcsin\dfrac{\|h\|}{\theta_r}\right] & \textbf{for } 0 \leq \|h\| \leq \theta_r \\ \theta_s & \textbf{for } \theta_r < \|h\| \end{cases}$$

Spherical

The semivariogram model is

$$\lambda(h;\theta)=\begin{cases} \theta_s\left[\dfrac{3}{2}\dfrac{\|h\|}{\theta_r}-\dfrac{1}{2}\left(\dfrac{\|h\|}{\theta_r}\right)^3\right] & \textbf{for } \mathbf{0} \leq \|h\| \leq \theta_r \\ \theta_s & \textbf{for } \theta_r < \|h\| \end{cases}$$

Tetraspherical

The semivariogram model is

$$\gamma(h;\theta)=$$

$$\begin{cases} \dfrac{2\theta_s}{\pi}\left(\arcsin\left(\dfrac{\|h\|}{\theta_r}\right)+\dfrac{\|h\|}{\theta_r}\sqrt{1-\left(\dfrac{\|h\|}{\theta_r}\right)^2}+\dfrac{2}{3}\dfrac{\|h\|}{\theta_r}\left(1-\left(\dfrac{\|h\|}{\theta_r}\right)^2\right)^{\frac{3}{2}}\right. \\ \theta_s \end{cases}$$

$$\textbf{for } 0 \leq \|h\| \leq \theta_r$$

$$\textbf{for } \theta_r < \|h\|$$

Pentaspherical

$$\gamma(h;\theta) = \{\theta_s \left[\frac{15}{8}\frac{\|h\|}{\theta_r} - \frac{5}{4}\left(\frac{\|h\|}{\theta_r}\right)^3 + \frac{3}{8}\left(\frac{\|h\|}{\theta_r}\right)^5 \right] \quad \textbf{for } 0 \le \|h\| \le \theta_r$$

$$\theta_s \quad \textbf{for } \theta_r < \|h\|$$

Exponential

$$\gamma(h;\theta) = \theta_s \left[1 - \exp\left(-\frac{3\|h\|}{\theta_r} \right) \right] \text{for all } \mathbf{h},$$

Gaussian

$$\gamma(h;\theta) = \theta_s \left[1 - \exp\left(-3\left(\frac{\|h\|}{\theta_r}\right)^2 \right) \right] \text{for all } \mathbf{h},$$

where $\theta_s \ge 0$ is the partial sill parameter and $\theta_r \ge 0$ is the range parameter. Because this model has unstable behavior without nugget, by default the Geostatistical Analyst adds a small nugget to the model, equal to 1/1000 of the sample variance computed for the data.

Appendix B

Radial Basis Functions

Multiquadric:

$$\varphi_1(r)=\sqrt{r^2}+c^2$$

Inverse multiquadric:

$$\varphi_2(r)=\frac{1}{\sqrt{r^2+c^2}}$$

Thin plate spline:

$$\varphi_3(r)=c^z r^z \ln(cr)$$

$$\varphi_3(r)=\left(c^2+r^2\right)\ln\left(c^2+r^2\right)$$

Multilog:

$$\varphi_4(r)=\ln\left(c^2+r^2\right)$$

Natural cubic spline:

$$\varphi_5(r)=\left(c^2+r^2\right)^{3/z}$$

Spline with tension:

$$\varphi_6(r)=\ln(cr/2)+\iota_0(cr)+\gamma$$

$$\iota_0(cr)=\sum_{i=0}^{\infty}\frac{(-1)^i(cr/2)^{zi}}{(i!)^z}$$

$$\phi_7(r)=\ln(cr/2)^z+E_1(cr)^2+\gamma$$

$$E_1(x)=\int_1^{\infty}\frac{e^{-tx}}{t}dt$$

Appendix C
WPS service “Cross validation” description

```
<?xml version="1.0" encoding="UTF-8"?>
<!--This example describes a the best semivarigram model -->
<wps:ProcessDescriptions xmlns:wps="http://www.opengis.net/wps/1.0.0" xmlns:ows="http://www.opengis.net/ows/1.1"
xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsDescribeProcess_response.xsd" xml:lang="en-US" service="WPS" version="1.0.0">
        <ProcessDescription wps:processVersion="2" statusSupported="true" storeSupported="true">
                <ows:Identifier>org.n52.wps.server.algorithm.interpolation.GeneralCV</ows:Identifier>
                <ows:Title>Cross validation of method selected</ows:Title>
                <ows:Abstract>Cross Validation</ows:Abstract>
                <ows:Metadata xlink:title="spatial" />
                <ows:Metadata xlink:title="geometry" />
                <ows:Metadata xlink:title="buffer" />
                <ows:Metadata xlink:title="GML" />
                <DataInputs>
                        <Input minOccurs="1" maxOccurs="1">
                                <ows:Identifier>Data</ows:Identifier>
                                <ows:Title>Points to be Interpolated</ows:Title>
                                <ows:Abstract>Points</ows:Abstract>
                                <ComplexData>
                                <Default>
                                        <Format>
                                                <MimeType>application/x-zipped-shp</MimeType>
                                        </Format>
                                </Default>
                                <Supported>
                                <Format>
                                        <MimeType>text/XML</MimeType>
                                        <Schema>http://schemas.opengis.net/gml/2.1.2/feature.xsd</Schema>
                                </Format>
                                <Format>
                                        <MimeType>application/x-zipped-shp</MimeType>
                                </Format>
                                </Supported>
                                </ComplexData>
                        </Input>
                        <Input minOccurs="1" maxOccurs="1">
                                <ows:Identifier>Field</ows:Identifier>
                                <ows:Title>Field with Z value</ows:Title>
                                <ows:Abstract>Value of Z</ows:Abstract>
                                <LiteralData>
                                <ows:DataType ows:reference="xs:string"></ows:DataType>
                                        <ows:AllowedValues>
                                                <ows:Value></ows:Value>
                                        </ows:AllowedValues>
                                </LiteralData>
                        </Input>
                        <Input minOccurs="1" maxOccurs="1">
                                <ows:Identifier>Method</ows:Identifier>
                                <ows:Title>Method to be evaluated (Kriging,IDW,Spline,KrigingUniversal)
</ows:Title>
                                <ows:Abstract>Write type method</ows:Abstract>
                                <LiteralData>
                                <ows:DataType ows:reference="xs:string"></ows:DataType>
                                        <ows:AllowedValues>
                                                <ows:Value></ows:Value>
                                        </ows:AllowedValues>
                                </LiteralData>
                        </Input>

                </DataInputs>
                <ProcessOutputs>
                        <Output >
                                <ows:Identifier>RMS</ows:Identifier>
                                <ows:Title>Root mean square deviation</ows:Title>
                                <ows:Abstract>Root mean square deviation</ows:Abstract>
                                <LiteralOutput>
                                <ows:DataType ows:reference="xs:double"></ows:DataType>
                                </LiteralOutput>
                        </Output>
                        <Output >
                                <ows:Identifier>StdRMS</ows:Identifier>
                                <ows:Title>Correlation coefficient</ows:Title>
                                <ows:Abstract>Correlation coefficient</ows:Abstract>
                                <LiteralOutput>
                                <ows:DataType ows:reference="xs:double"></ows:DataType>
                                </LiteralOutput>
                        </Output>
                        <Output >
                                <ows:Identifier>Best_Param</ows:Identifier>
                                <ows:Title>Equation of Regression</ows:Title>
                                <ows:Abstract>Regression</ows:Abstract>
                                <LiteralOutput>
                                <ows:DataType ows:reference="xs:string"></ows:DataType>
                                </LiteralOutput>
                        </Output>
                        <Output >
                                <ows:Identifier>GraphCross</ows:Identifier>
                                <ows:Title>Link cross validation graph</ows:Title>
                                <ows:Abstract>Cross validation graph</ows:Abstract>
                                <LiteralOutput>
                                  <ows:DataType ows:reference="xs:string"></ows:DataType>
                                </LiteralOutput>

                        </Output>
                        <Output >
                                <ows:Identifier>GraphFit</ows:Identifier>
                                <ows:Title>Link Fit best parameter</ows:Title>
                                <ows:Abstract>Link Fit best parameter</ows:Abstract>
                                <LiteralOutput>
                                  <ows:DataType ows:reference="xs:string"></ows:DataType>
                                </LiteralOutput>
                        </Output>
                        <Output >
                                <ows:Identifier>Iterations</ows:Identifier>
                                <ows:Title>Iterations</ows:Title>
                                <ows:Abstract>Iterations</ows:Abstract>
                                <LiteralOutput>
                                  <ows:DataType ows:reference="xs:string"></ows:DataType>
                                </LiteralOutput>
                        </Output>
                </ProcessOutputs>
            </ProcessDescription>
</wps:ProcessDescriptions>
```

Appendix D

WPS service “Interpolate” description

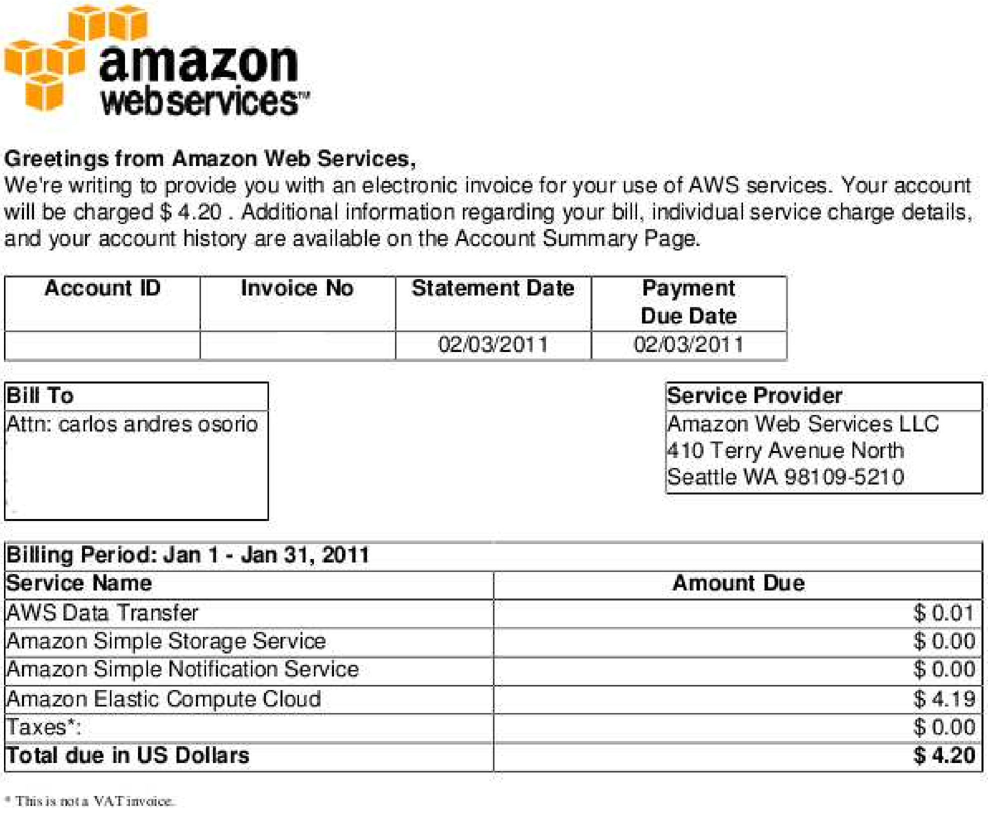
```
<?xml version="1.0" encoding="UTF-8"?>
<!--This example describes a Kriging -->
<wps:ProcessDescriptions xmlns:wps="http://www.opengis.net/wps/1.0.0" xmlns:ows="http://www.opengis.net/ows/1.1"
xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsDescribeProcess_response.xsd" xml:lang="en-US" service="WPS" version="1.0.0">
	<ProcessDescription wps:processVersion="2" statusSupported="true" storeSupported="true">
		<ows:Identifier>org.n52.wps.server.algorithm.interpolation.interpolationDivZ</ows:Identifier>
		<ows:Title>Interpolate data using Kriging, IDW, KrigingUnivversal or Spline</ows:Title>
		<ows:Abstract>IDW;pow;num--Kriging;model;range;sill;nugget;numOfpointsSearch--;
Spline;model;factor;num</ows:Abstract>
		<ows:Metadata xlink:title="spatial" />
		<ows:Metadata xlink:title="geometry" />
		<ows:Metadata xlink:title="buffer" />
		<ows:Metadata xlink:title="GML" />
		<DataInputs>
			<Input minOccurs="1" maxOccurs="1">
				<ows:Identifier>Data</ows:Identifier>
				<ows:Title>Points to be Interpolated</ows:Title>
				<ows:Abstract>Points</ows:Abstract>
								<ComplexData>
				<Default>
					<Format>
						<MimeType>application/x-zipped-shp</MimeType>
					</Format>
				</Default>
				<Supported>
				<Format>
					<MimeType>text/XML</MimeType>
					<Schema>http://schemas.opengis.net/gml/2.1.2/feature.xsd</Schema>
				</Format>
				<Format>
					<MimeType>application/x-zipped-shp</MimeType>
				</Format>
				</Supported>
				</ComplexData>
			</Input>
			<Input minOccurs="1" maxOccurs="1">
				<ows:Identifier>Field</ows:Identifier>
				<ows:Title>Field with Z value</ows:Title>
				<ows:Abstract>Value of Z</ows:Abstract>
				<LiteralData>
				<ows:DataType ows:reference="xs:string"></ows:DataType>
					<ows:AllowedValues>
						<ows:Value></ows:Value>
					</ows:AllowedValues>
				</LiteralData>
			</Input>

			<Input minOccurs="1" maxOccurs="1">
				<ows:Identifier>Method</ows:Identifier>
				<ows:Title>Method with parameters </ows:Title>
				<ows:Abstract>Method with param</ows:Abstract>
				<LiteralData>
				<ows:DataType ows:reference="xs:string"></ows:DataType>
					<ows:AllowedValues>
						<ows:Value></ows:Value>
					</ows:AllowedValues>
				</LiteralData>
			</Input>
			<Input minOccurs="1" maxOccurs="1">
				<ows:Identifier>Resolution</ows:Identifier>
				<ows:Title>Resolution</ows:Title>
				<ows:Abstract>Resolution</ows:Abstract>
				<LiteralData>
				<ows:DataType ows:reference="xs:double"></ows:DataType>
					<ows:AllowedValues>
```

```
                                          <ows:Value></ows:Value>
                                   </ows:AllowedValues>
                            </LiteralData>
                     </Input>
              </DataInputs>
              <ProcessOutputs>
    <Output>
       <ows:Identifier>Result</ows:Identifier>
                     <ows:Title>Result</ows:Title>
                     <ows:Abstract>Result</ows:Abstract>
               <ComplexOutput>
                     <Default>
       <Format><MimeType>application/WMS</MimeType><Encoding>UTF-8</Encoding></Format>
      </Default>
                     <Supported>
              <Format><MimeType>image/tiff</MimeType><Encoding>UTF-8</Encoding></Format>
              <Format><MimeType>image/geotiff</MimeType><Encoding>UTF-8</Encoding></Format>
              <Format><MimeType>application/WCS</MimeType><Encoding>UTF-8</Encoding></Format>
                     </Supported>
         </ComplexOutput>
       </Output>
              <Output>
                            <ows:Identifier>Duration</ows:Identifier>
                            <ows:Title>Duration</ows:Title>
                            <ows:Abstract>Duration</ows:Abstract>
                            <LiteralOutput>
                              <ows:DataType ows:reference="xs:string"></ows:DataType>
                            </LiteralOutput>
                     </Output>
              </ProcessOutputs>
       </ProcessDescription>
</wps:ProcessDescriptions>
```

Appendix E

Cost of services used in Amazon AWS



Greetings from Amazon Web Services,

We're writing to provide you with an electronic invoice for your use of AWS services. Your account will be charged $ 4.20 . Additional information regarding your bill, individual service charge details, and your account history are available on the Account Summary Page.

Account ID	Invoice No	Statement Date	Payment Due Date
		02/03/2011	02/03/2011

Bill To
Attn: carlos andres osorio

Service Provider
Amazon Web Services LLC 410 Terry Avenue North Seattle WA 98109-5210

Billing Period: Jan 1 - Jan 31, 2011	
Service Name	**Amount Due**
AWS Data Transfer	$ 0.01
Amazon Simple Storage Service	$ 0.00
Amazon Simple Notification Service	$ 0.00
Amazon Elastic Compute Cloud	$ 4.19
Taxes*:	$ 0.00
Total due in US Dollars	**$ 4.20**

* This is not a VAT invoice.

Masters
Program
in **Geospatial**
Technologies

Parallelization of Web Processing Services on Cloud Computing: A case study of Geostatistical Methods

Carlos Andres Osorio Murillo

Dissertation submitted in partial fulfilment of the requirements for the Degree of *Master of Science in Geospatial Technologies*

UNIVERSIDADE NOVA ISEGI

UNIVERSITAT JAUME·I

WESTFÄLISCHE WILHELMS-UNIVERSITÄT MÜNSTER

Masters
Program
in **Geospatial Technologies**

Supported by:

Education and Culture

ERASMUS MUNDUS