Community Based Open Source Geographical Classical Data Analysis

Jayapandian N, Madhavendra Singh Negi, Samridh Agarwal, Ajay Prasanna

Abstract: The traditional Geographical Information Systems (GIS) have to be migrated to the internet eventually much like every other software today. The article has explored ways of utilizing the Open Geospatial Consortium (OGC) standards to come up with ways of achieving a workflow for the development of a service-based implementation of a customized Web Processing Service (WPS). The proposed concept has explored multiple workflows using various combinations of the publishing and development options and the simplest and the least resource intensive one has been identified as the outcome of this project. The workflow identified was then split into two section to make it even more simplify and adaptable, aiding development from the WPS that has to publish. The development process used for the final workflow is done without the use of a resource intensive IDE keeping in mind the major aim of the proposed model is to reduce the dependency on resource intensive software and services. The proposed model is built solely on open source platforms which are in tandem with second stipulation of proposed model is promoting community-based development. The proposed system provides the better execution time and retrieval time. The execution time is compared with similar system, open source Geographical system provide less execution time. The retrieval time is also reduced this indicated Quality of Service is increased.

Index Terms: Data Collection, Geographical Information System, Open Source, Road Map Analysis, Web Processing Service, Workflow.

I. INTRODUCTION

This model is tested multiple methods for development of a WPS using geotools which is a java library that was used for creating the project. Some functionality were used from the 520 North community archives, additionally future component is planned where in 520 North WPS clients will be used to give a WPS service in place of the WPS builder. 520 North WPS clients act as the go between framework for processing over an in-house or a third-party server. The objective of this project was to find a simplistic workflow for development of a WPS as well as publishing it as a service. The need for WPS in GIS is gaining importance due to the advent of cloud-based services as well as distributed computing. This is important as it brings along with it the promise easier collaboration as well as faster and unrestricted development options. This system however is bound to give rise to heterogeneity in the type of data that the system can process and to handle that the GeoServer has been employed [1]. Open source technology is gaining great popularity as more and more organizations are adopting it due to the multitude of benefits that it carries, it is becoming a part of the computing strategy. The proposed system workflow is modeled keeping in mind that the system will utilize only open source resources for the implementation of the workflow. The various components are gone into the implementation of the workflow [2]. GeoServer is developed on java as a software server which plays a major role in permitting the user to work on the server just as they would work on a desktop based GIS. Majorly designed with the characteristics of interoperability, it publishes the data using standard and the source of data can be from any major spatial data source. GeoServer has developed as a method for simpler publishing as well as visualizing geospatial data, enabling the user to view complex shapefiles over the server. What makes it possible for GeoServer to have a request and response model as specified under the OGC, is that it uses the Spring Framework. This enables the system to give the users an API built on the spring-mvc-framework; in case of the GeoServer the API is a REST API [3]. GeoServer provides capabilities that allow for development of large number of functionalities one such is the WPS request builder is an extension that enables the developed WPS to be executed. To do this the administrator interface is used and under the demo uses the builder can be found for the same. The builder is used to demonstrate processes, and construct the examples of the given requirement. For using the request builder, the selection box appears with listing of all the available process additionally there are two buttons one of which is to submit the requests of the WPS requests and the second one displays the POST request looks like when used, these are the functionalities WPS Request Builder primarily. Dependence on the input and process selected leads to the change in display [4]. Apache maven was developed in order to help in packaging JAVA projects into project jar files, essentially serving as extremely effective project management. In development maven contributes by allowing seamless integration external packages and libraries. This is done by tapping into maven repositories while development of services, the external resources used in the artifact that are not available in the local repository are automatically downloaded, configured and integrated by either an integrated development environment (IDE) or can be done through systems command like “mvn clean install”.

Revised Manuscript Received on May 18, 2019

Jayapandian N. Department of Computer Science and Engineering, Christ University, Bangalore, India.
Madhavendra Singh. Department of Computer Science and Engineering, Christ University, Bangalore, India.
Samridh Agarwal. Department of Computer Science and Engineering, Christ University, Bangalore, India.
Ajay Prasanna Y. Department of Computer Science and Engineering, Christ University, Bangalore, India.

Published By:
Blue Eyes Intelligence Engineering & Sciences Publication
II. RELATED WORK

The GIS based systems that are available are mostly proprietary or are desktop based this causes an issue given that the modern development and even consumption environment has been gravitating towards a service based open sourced solution for most of the major products [6]. So, it would be a smart move to expedite this process, however is not an easy process. Due to the complicity of a GIS in terms of the way the data is stored to the way it is processed and displayed, the migration to a full-fledged web GIS can be a cumbersome task. Hence to make a workflow that can be used as a simple way of developing a WPS and then publishing it has to be made the project deals with creation of such a workflow [7]. Additionally, to handle the data heterogeneity as well as the issues pertaining to visually presenting it is also dealt with as a part of the project. The major outcome of the proposed system is to identify a workflow that can aptly fulfill the aforementioned requirements in the simplest possible way while using open source components to do it. This GIS is also used in smart city device to manage device location [8]. Handling larger GIS data is facing data storage problem, these kinds of problems to solve MLEwCE algorithm [9]. This data is stored in cloud server, online cloud data storage is also facing scheduling problem, this problem is to solve backfilling algorithm [10].

III. PROPOSED METHODOLOGY

The system workflow has been kept minimalistic; also the workflow is entirely implemented on open source platforms. The implementation is been done is stages and the stages include the data publication and the data processing phase. For both the stages the system relies a lot on the GeoServer in terms of backend support as the server provides an environment that can be used to carry out various actions that let users work on various GIS features. For instance, the user may use the map service and then publish a service of a map application alternatively the publishing of maps graphical form is also possible [11]. This can effectively convert a shapefile to an image file, or any other convenient format that one needs without the use of any traditional desktop software like ArcGIS or QGIS. This feature added to the availability of features like WPS build makes the GeoServer apt for the workflow [12]. So, the initial setup of the project begins with the configuration of the GeoServer, this is a simple process for which three components are needed namely the Apache Tomcat, GeoServer and the WPS extension for the GeoServer that is available in the archives of the Open Source Geospatial Foundation [13].
From the given workflow, it explains about the how the process is selected according to the process requested and then the result generation take place with validation of it which is split into two parts: storing of result and rejection of output with a display error and after the result is displayed, the whole process is terminated as the final step. This workflow makes it possible for the WPS to process data and display outputs. The figure 2 explains the process of uploading and publishing of data in GeoServer which is done with adding data to new stores and then the data file uploading request is sent to the server which helps in data upload with particular configuration and ends with publishing of the data according to configuration and terminating the whole process. The above two in tandem form what can be considered the architecture of the system if it were being analyzed according to a traditional software engineering model for development.

IV. RESULT AND DISCUSSION

The major decision is the choice between possible system implementations that can deliver the system requirement. After much analysis of the subject there were three major contenders for the model and they are given as follows.

WPS plugin builder on Quantum GIS, It is the QGIS is an open source desktop based GIS software tool that has come up a popular alternative to ArcGIS and GRASS GIS, it comes with the plugin builder option that can be used to develop WPS service by setting up the plugin builder. This implementation needs the QGIS to interface with the WPS client over a server, after the server connection is made the WPS services can be accessed. After the setup and configurations are made the user can access the predefined as well as custom made WPS that are uploaded to the server and the process is executed by QGIS, the output can be viewed directly on the QGIS output prompt. This allows the configuration of multiple servers that can run a WPS for the user, the issue with this system is that QGIS is considerably more difficulty to use and has a labeling interface that is fairly complicated. Which is why is not considered for the implementation of the workflow of the project.

WPS request builder on GeoServer, It is the WPS request builder is a demo feature that GeoServer has rolled out with the GeoServer 2.4, this has enabled the developers to create project in java using various maven repositories and then using the clean build command package it into a jar file that can be directly pasted in to the lib section in the GeoServer server webapp directory. On the next server restart the WPS is added to the WPS request builder and the functionality can be access after the GeoServer login through the demo builder feature. This is the method that has been chosen of the implementation of the workflow for the simple reason that amongst the three it is only one that allows the development of a fully-fledged web based GIS and is thereby the only method that meets the goals of the project. Additionally, it must be observed that this implementation is the simplest due to the lack of external dependences on external software like the QGIS.
Framework based on WAMP and QGIS, it is the WAMP based framework would entail the use of a three tiered architecture that would use WAMP in combination with ALOV to form the logic tire while the presentation tier would be made by the user of various PHP scripts and the data layer or tire would in turn use a Relational DataBase Management System (RDBMS) like PostgreSQL or MySQL. On close inspection PostgreSQL would be a more suitable choice at is lends some spatial support in the form of the PostGIS which is a tailor-made database extension for the PostgreSQL and allows the queries to cater to the needs to geospatial data. Thus, it would be the preferred RDBMS for a system such as the one the project wishes to emulate. However this implementation would not be able to fulfill all the system requirements as without the JSP that can be used when using the Apache Tomcat the system will not be able to directly implement the entire GIS and due to this the Tomcat based implementation has been picked for the implementation of the workflow. The figure 3 is an output of the buffer feature collection WPS on a sample data set obtained from WorldMap, which is developed by the Center for Geographic Analysis at Harvard University. This is how WPS outputs look like if they are generated in the Web Feature Service (WFS) format, which is another OGC standard. Similar WPS can be developed and they can be employed by analysis and processing of geospatial data over the web by the use of workflows like this one.

This workflow is apt for a local server based implementation, however it can be adapted to the requirements of a larger organization or a community. The other part of the workflow is publishing the data to GeoServer, thereby lending it a certain level of homogeneity as it possible to convert this data to multiple formats. The GeoServer makes it possible for the data to be converted to the formats that are visible above this makes it a unique system that can help the interfacing of multiple application for various users simultaneously. The GeoServer also allows the creation of users on the server with different roles, additionally the roles in themselves may not be restrictive the administrator of the server can define new rules that match up to the requirement of the system. The figure 4 is obtained for a free repository of roads and for classical geographical analysis of this kind of data it would need for a researcher to install and learn how to use a desktop based GIS application or system. The proposed system will however allow the user to publish the data or have an administrator of the server publish it and the user can then directly use the GeoServer “Layer Preview” option to view the file in any format that is needed. The use OpenLayer also allows for browser-based retrieval of the data associated to the visual data of the spatial files, so the point data can be displayed on clicking the associated visual data. The comparative analysis made against a few identified systems has rated the system as is described in the following comparative performance analysis.

<table>
<thead>
<tr>
<th>Retrieval Time Comparison</th>
</tr>
</thead>
<tbody>
<tr>
<td>Indian_Water (ms)</td>
</tr>
<tr>
<td>19.8</td>
</tr>
</tbody>
</table>

The figure 5 datasets used has been taken from the DIVA-GIS open repository of country wise data, the analysis goes on to show that the proposed system can compete with the preexisting desktop-based systems.
Table 2. Execution Time Comparison

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Indian_Water (ms)</th>
<th>ARG_ROADS (ms)</th>
<th>Rus_Rail (ms)</th>
<th>AUS_ADMIN (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>QGIS</td>
<td>7.6</td>
<td>8</td>
<td>8.2</td>
<td>7.4</td>
</tr>
<tr>
<td>SAGA GIS</td>
<td>8.2</td>
<td>8.6</td>
<td>8.6</td>
<td>8.2</td>
</tr>
<tr>
<td>Proposed System</td>
<td>8</td>
<td>8.4</td>
<td>8.6</td>
<td>8.2</td>
</tr>
</tbody>
</table>

Table 3. Shared Memory Comparison

<table>
<thead>
<tr>
<th>System</th>
<th>System Memory (Mb)</th>
</tr>
</thead>
<tbody>
<tr>
<td>QGIS</td>
<td>221 MB</td>
</tr>
<tr>
<td>SAGA GIS</td>
<td>61 MB</td>
</tr>
<tr>
<td>Proposed System</td>
<td>39 MB</td>
</tr>
</tbody>
</table>

The figure 6 parameter used for the comparison of our system with the desktop GIS, in perspective of the user is the system memory. The proposed system here outdoes the other systems due to a browser-based implementation and the user system is not as burdened. Additionally, there can be further improvements in the proposed system as is discussed in the later section of the paper to improve the performance of the system.

V. CONCLUSION

The proposed method is successfully tested the workflow for publishing and using WPS over GeoServer additionally the project has also demonstrated that data standardization can be done by publishing data to the GeoServer before processing. The proposed system has tested GeoServer to be a successful way of ensuring interoperability for this workflow as it allows the data to be converted to multiple files which allows it to be used elsewhere in whatever format is needed. The proposed method has identified the Apache Tomcat in combination with GeoServer to be the most effective workflow for the creation of a full-fledged GIS system that can successfully compete with any desktop based GIS.

VI. FUTURE SCOPE

The workflow in itself is a fairly refined one and while all workflows can be refined, the major change is needed in the way the WPS is used by the end user. The WPS request builder is a good tool however is not as intuitive and user friendly as one would like a GIS tool to be, so the system can be improved by adding a WPS client like the 520 North WPS client that would by and large make the user experience better and the developer experience easier. The system also has little to no control on the WPS overwhelming the server it is running on and there needs to be additional functionalities put in to prevent the system from overloading. One of the available solutions to this problem is to change the GeoServer configuration under the WPS security options, but this is not a very efficient solution. So, by adding new constraints like limiting the size of the input or by finding new stopping conditions that optimize the resource consumption of a WPS the system resources may be conserved and the system overload prevented.

REFERENCES


AUTHORS PROFILE

Dr. Jayapandian N is Currently Working as an Assistant professor at the CHRIST (Deemed to be University), Department of Computer Science and Engineering, Bangalore, India. His research interest includes information security, cloud computing, and grid computing. Dr. Jayapandian is completed Ph.D (Information & Communication Engineering) from Anna University. He holds a Bachelor of Technology degree in Information Technology from Anna University and Master Degree in Computer Science and Engineering from Anna University. He is published various research article in reputed international journals. He is an active reviewer of reputed international journals. He has participated in numerous national and international conferences and has made a remarkable contribution to cloud data security field and publishing several articles. He is published various book chapters under the concept of data security.

Mr. Madhavendra Singh Negi is a recent graduate from the Faculty of Engineering at CHIRST (Deemed to be University), having secured a Bachelor of Technology in Computer Science and Engineering. He is worked extensively in software development and data analytics by way of carrying out a varied number of projects in a number of domains. He has been interest in geographic information systems and has worked on integration of GIS based features into BI systems.

Mr. Ajay Prasanna holds a Bachelor of Technology degree in Information Technology from CHIRST (Deemed to be University). He is sincere and hard working with a tremendous drive to fulfill his tasks and also obtained a good recognition from the whole department. He is a keen observer and has the requisite aptitude to take up any difficult assignment in his field of study. He has really good computational skills. He was involved and excelled in departmental activities. His research interest includes information security, cloud computing and mobile computing. He has published a research article in a reputed international journal.

Mr. Samridh Agarwal is a recent graduate from the Faculty of Engineering at CHIRST (Deemed to be University), having secured a Bachelor of Technology in Computer Science and Engineering. He is worked in data analysis and machine learning field. He has been interest in GIS data analysis and data management field.