A Geospatial Orchestration Framework on Cloud for Processing User Queries

Jaydeep Das*, Arindam Dasgupta†, Soumya K. Ghosh‡, Rajkumar Buyya§

*†Advanced Technology Development Centre
Indian Institute of Technology Kharagpur, India

‡Department of Computer Science and Engineering
Indian Institute of Technology Kharagpur, India

§Cloud Computing and Distributed Systems (CLOUDS) Lab
Department of Computing and Information Systems
The University of Melbourne, Australia

E-mail: jaydeep@iitkgp.ac.in*, adgkgp@gmail.com†, skg@iitkgp.ac.in‡, rbuyya@unimelb.edu.au§

***Abstract*—The demand for computing resources to process the geographical information (GI) queries has been increased drastically. The query helps the users to get the variety of information to serve their needs. Resolving the spatial queries, huge number of heterogeneous data sources along with different computing services are involved. Getting appropriate results within a specific time bound, orchestration among those data sources and web services are required. These services are available on the web and require different resource specifications in order to resolve a geospatial query. A cloud infrastructure has been utilized for scalable resource allocation. An orchestration engine has been developed to access the geospatial resources according to query requirement. In this paper, we have proposed and developed geographical data query processing framework which orchestrates spatial services according to user query in cloud environment. The empirical experimentation shows the efficiency of the proposed framework to resolve spatial queries in timely manner.**

Keywords: Spatial Query, Orchestration, Spatial Web Service, Geospatial Cloud

I. Introduction

The demand for geospatial data has been increased with the development of data acquisition systems and various geographical information system (GIS) software. With the availability of such software people expect variety of geographical information (GI) instantly for their daily use. However, responding to these queries in timely manner require not only a large memory space, but also huge computing power. The need of such computing resources depends on the user query. These resources can be either memory space or data processing speed. Specifically, when the GIS information are fetched in mobile devices. Again, the processing power and memory space requirement varies with the type of incoming query from the user. In this situation, the demand of processing power and memory requirement depends on the amount of data resources and the number of computing functions involve with the user query. Hence, demand of computing power and memory varies extremely. A framework is needed to provide computing resources on demand. In such situation, cloud computing environment provides the most efficient trade-off between the GIS query and system resources.

Spatial cloud computing [1] refers to the cloud computing paradigm that is driven by geospatial sciences, and optimized by spatiotemporal principles for enabling geospatial science discoveries and cloud computing within distributed computing environment. Spatiotemporal principles [2] are critical to enable the discover-ability, accessibility and usability of the distributed, heterogeneous and massive data. For computation intensive problem, cloud computing select the resources such that it optimize the utilization of high end computation resources. It enables the timely response either worldwide or local users through geospatial optimization. Further, it also assists the design of spatiotemporal data structure and algorithms to optimize the information workflow in order to solve complex problems. Again, it needs multiple processing of geospatial data from multiple heterogenious data sources. The geoprocessing functions in cloud environment can bring scalable, on-demand, and cost-effective services. Yue et al. [3] compared different geoprocessing services in different public cloud computing platforms. More complexities are involved in the case of heterogeneous data resources. Web feature services (WFS) accumulate these heterogeneous data into a single platform to resolve the user queries [4]. Further, it is processed by web processing services (WPS) module as per the user query requirements and demands high amount of RAM. Scaling up or scaling down of RAM requirement is very much needed for processing service. It can be done by assigning virtual machine (VM) in cloud environment.

A spatial query can be split into several small query trees. From these trees, the orchestration engine chooses the optimal one using business logic [5]. Selected query tree can consider as a complex spatial data analysis task. Workflow management helps in realization of parallel implementation of the spatial analysis tasks.

In this work, the complex spatial queries are considered. The resolution of the query involves multiple heterogeneous geospatial data resources. In order to resolve such complex spatial queries, the following issues are considered in this work.

- Geospatial data are voluminous and distributed over multiple data centers at different locations.
- Query execution is complex because data are coming from multiple sources with different data format.

To resolve user query, a sequence of geospatial web services is needed. Sequencing of existing services is achieved by a complex information service framework mentioned in [6]. Bernard et al. [7] have developed a Web service framework for heterogeneous environmental information systems.

The rest of the paper is organized as follows. Section II presents the background of the work. Section III gives an overview of our proposed system architecture. Section IV presents the methodology of our system. Section V elaborates a case study and results. Finally, Section VI concludes the paper and discusses the future works.

II. Background

In this section, we present some related terminologies and technologies which are utilized in our work.

A. Geospatial Web Services

Geospatial web services are Open Geospatial Consortium (OGC) standards based mechanisms for electronically connecting to customers, suppliers and partners focusing on geospatial information. Geospatial web services emphasis three main purposes - data discovery, data visualization and data access.

Types of geospatial services are as follows -

- *Web Features Services (WFS)* [8] is interfaces for defining data handling operations like create, update, delete a geographic feature instance.
- *Web Processing Service (WPS)* [9] provides a platform with different geo-processing functions. It could access across the network to utilize a preprogrammed computation model that operates on spatially referenced data.
- *Web Map Service (WMS)* returns one or more georegistered map images from distributed geospatial databases in JPEG, PNG, etc. format on the response of WMS request.
- *Catalog Services for the Web (CSW)* is needed to publish the geospatial characteristics and search group of metadata for data, services, and related information objects provided by different sources. So, the client system can automatically bind with the required geospatial services. CSW interface helps a client to query on catalogs to discover resources. A CSW has different requirements for its three main types of users - Resource User, Resource Provider and Registry Manager.

B. Workflow

Workflow is a sequence of computational and data processing tasks. Workflow technologies are frequently used for complex analysis of engineering, business, or scientific processes. For on-demand complex data analytics in the cloud environment, a service-oriented workflow architecture [10] is needed.

C. Orchestration Engine

Orchestration is the description of communications, and messages flow between services in the context of a business process [11]. Goal of orchestration is to make participation of web services across the enterprise boundary to access large information. A geospatial orchestration engine, embedded with a rule repository, has been proposed in paper [6]. This orchestration engine composes different geospatial web services across the enterprise boundary.

D. Spatial Query Parsing

Parsing is a stage in the processing of a query statement. Issuing a spatial query statement, an application makes a parse call to the spatial database. Parsing should be done in such a manner that accuracy of the result is high within a short time span considering I/O cost more than CPU cost. Unlike relational database query, spatial query deals with extremely large volumes of complex objects with spatial extension [12].

III. System Architecture

A multilayer client-server geospatial cloud system architecture, as shown in Fig. 1, is customized to serve our purpose such as scalable computation and suitable VM assignment. It illustrates an implementation scenario with selected open source software components, such as GeoServer, ArcGIS, GRASS GIS, etc., supporting OGC web services standards interface in cloud environment. There are three layers in this architecture- Client, Application and Data layer where Application and Data layers are in cloud [13]. Clients(thick and thin) in Client layer can access applications and data using request-response method from cloud. Clients are able to view resultant maps according to their spatial query. The Application layer helps communicating between clients and data providers. On top of this layer, a web server presenting web services(catalog/process/map) and serving requests to and response from application servers considered as the access point of the system. Application services implement CSW, WFS, WCS and WPS.

- Catalog server applications keep track of metadata information about data and processes, received from different sources. This step becomes quite essential due to the large amount of spatial data in the Cloud. Furthermore, a well-defined approach following the publish-find-bind service framework is defined in the OGC Web Services architecture.
- Data server applications which are used to provide spatial data to the users, categorized in standardized service

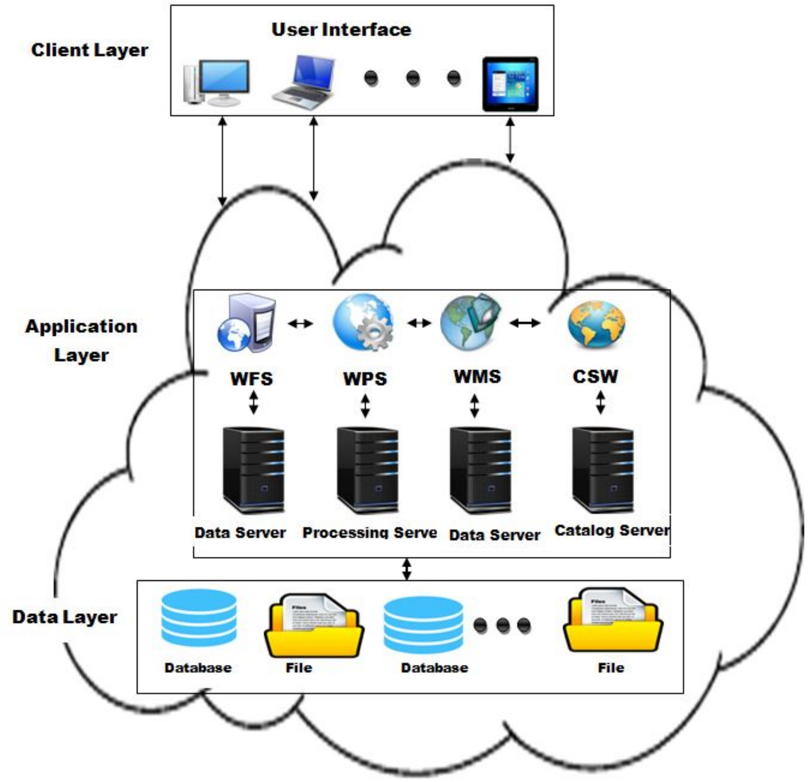

Fig. 1. Spatial Cloud System Architecture

forms, like WMS for map images, WFS for vector data and web coverage services (WCS) for grid data.

- Processing server applications offer a repository of geospatial processes and allow users to apply them over spatial data by implementing WPS standard. Users may query about details of every process, provide the processing service along with these parameters, define a certain bounding box, and provide data having complex values such as binary data and XML structures. Input data given by the user will further be modified by processing units. These units can either be newly developed ones or existing GIS software tools (e.g., Grass GIS). An internal communication interface between the processing server, and the processing unit is required. This is introduced in terms of a unified modeling language (UML) sequence diagram which implements the WPS request handler component.

All the spatial data and information are available in the Data layer. This layer is used to retrieve spatial data and provide various standardized services for further computations. File systems, database management systems of different national and international organizations are accessible from this layer.

IV. Methodology

A workflow model has been developed based on OGC standard geospatial services. This workflow model (see Fig. 2) is utilized to generate derived information by accessing different geospatial services according to user query. The user query is interpreted by business logic in the orchestration engine. After parsing the user query, a query tree will be generated by considering the essential geospatial services. It is the responsibility of the orchestration engine to map the query tree into the workflow model. According to this workflow model, different web services will be required to access and list of data sources and processing sources will generate. To achieve this, the orchestration engine will communicate with registry service. After getting the information about different web services and sources, orchestration engine communicates(binds) with virtual machines, which provide such web services and data sources. Maintaining the sequence (or parallel) of accessing web services and data sources, according to workflow, is a big challenge. It needs a cloud environment which can perceive, reason, learn GI services, and apply these services intelligently to construct the workflow of user query [14]. Web services are not available in a single virtual machine, rather these services are distributed in many virtual machines in a cloud platform. So, in cloud, parallel web service access can be possible. After processing of web services and data sources, results get back to orchestration engine and produce the query response to the user. User query can break into some specific service pattern

$$\textit{select } S_F \textit{ from } S_D \textit{ where } S_C$$

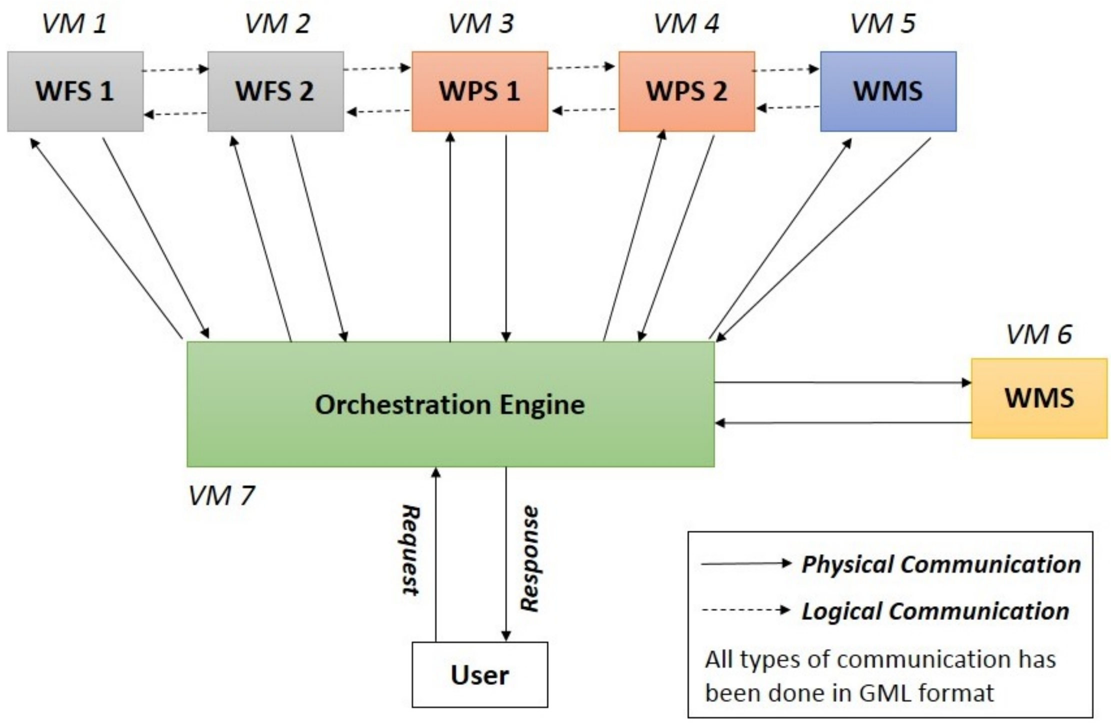

Fig. 2. Workflow in Spatial Cloud System

- Let S_F be a collection of feature services available in the cloud in form of WFS, denoted as $S_F =< S_{F_1}, S_{F_2}, \cdots, S_{F_n} >$.
- Let S_D be a collection of data services available, denoted as $S_D = < S_{D_1}, S_{D_2}, \cdots, S_{D_n} >$.
- S_C is the query predicate which depends on the business logic of orchestration engine and based on the logic different WPS services are called and let S_P be a collection of processing services available in the cloud in the form of WPS, denoted as $S_P = < S_{P_1}, S_{P_2}, \cdots, S_{P_n} >$.

We have represented different service flows in a sequence diagram which is presented in a Fig. 3.

V. Case study

In this work, the spatial data set (Land Use Land Cover and Road) of Purulia and Hatasuria (Bankura), West Bengal, India are considered for generating workflow according to user query. The spatial reference system, EPSG:32645, is used for displaying various maps.

A. Query resolution

Query from the user: *Find the suitable(top 6) places within Purulia and Hatasuria (Bankura), West Bengal, which has at least 50 acres industrial lands and the distance from the high road less than 1 kilometer.*
Query resolutions are as follows:

SELECT area_name
FROM Purulia
WHERE area $\geq$ 50 and road = 'High Road' and Overlap (road.shape, Buffer (area.shape, 1))
ORDER BY area desc;

SELECT area_name
FROM Hatasuria
WHERE area $\geq$ 50 and road = 'High Road' and Overlap (road.shape, Buffer (area.shape, 1))
ORDER BY area desc;

In order to resolve these queries, a predefined workflow is needed for the orchestration engine. In this work, the workflow to solve such type of queries has been developed. After getting the user request, the parser interprets the query string and identifies the relevant geospatial services to solve the query. The predefined workflow model is mapped with the related services and produces a service chain. Then the service will be executed by the orchestration engine to produce the result.

The steps for generating workflow are as follows:

1) Filter out the lands which have land area at least 50 acres using WFS *getFeature* service.
2) Create 1 km buffer of each filtered area using WPS *BufferFeatureCollection* service.
3) Filter out the specific roads from the road database using WFS *getFeature* service.
4) Make intersection lands buffer with filtered roads using WPS *IntersectionFeatureCollection* service.
5) Filter intersected lands using WPS *IntersectionFeatureCollection* service.

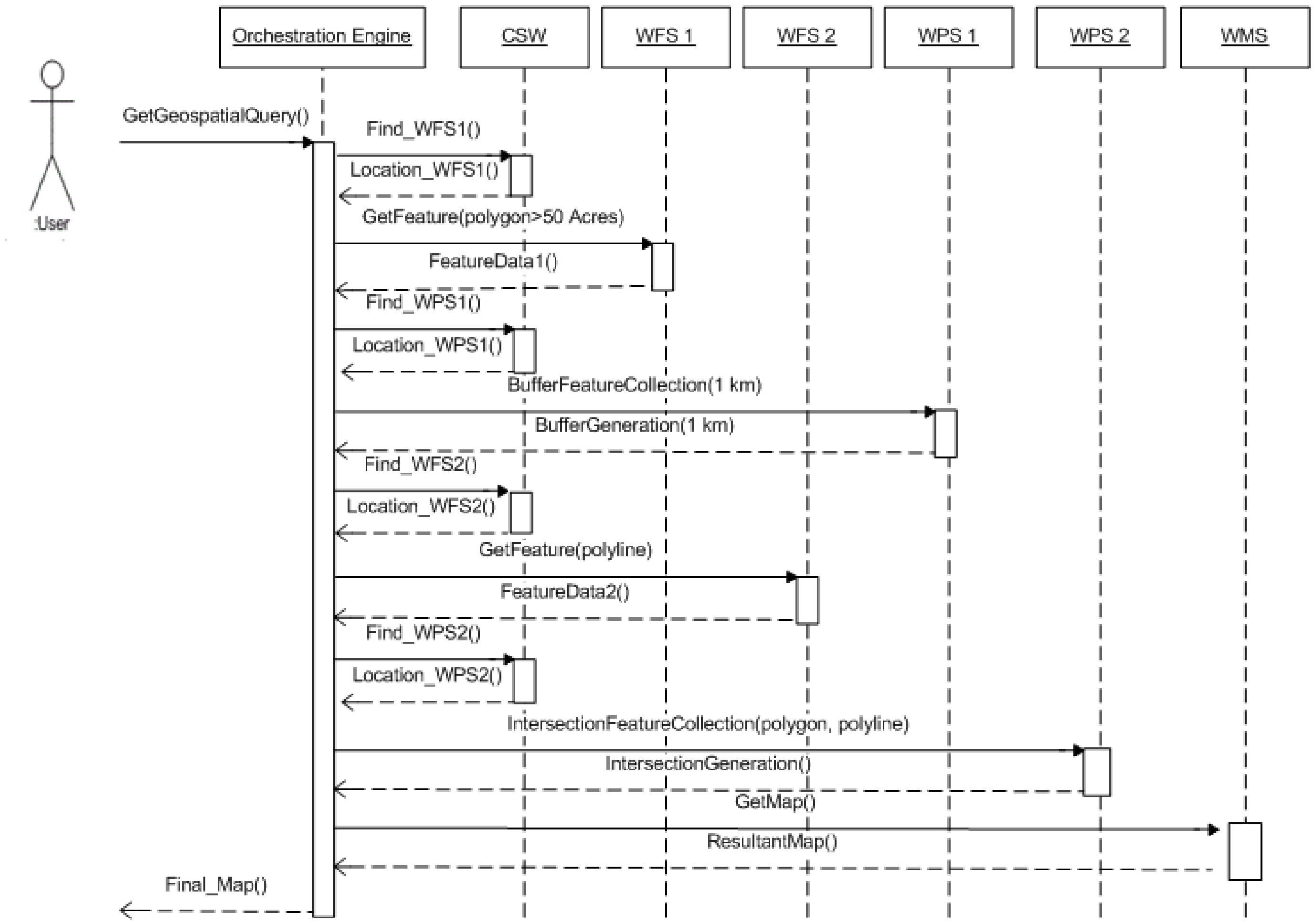

Fig. 3. Sequential accessing of OGC web services for geospatial user query

6) Generate geography markup language(GML) of the ascending order lists of resultant lands with *getFeature* service.

From both ascending order lists, user can choose suitable places meeting his requirements (land and road) from Purulia and Hatasuria. From fig. 4, we can observe that two parallel flow execute in a distributed system. Cloud computing is appropriate environment for executing these kinds of parallel operations in timely manner. Web services i.e., WFS, WPS and WMS are called several times. If these services are available in different virtual machines, then executions of jobs are done in short time span. However problem may occur to synchronize results.

B. Experimentation

To illustrate operational flow, we have taken snapshots of each step of the workflow. These are shown step wise in the figure Fig. 5. Fig. 5a shows all the areas of Purulia. Next Fig. 5b shows the filtration result of the industrial areas of Purulia. After creation of buffer of 1 km, the industrial areas look like the one shown in Fig. 5c. Similarly, road network of Purulia is shown in Fig. 6a. From this data high roads are filtered and is shown in Fig. 6b. The resultant intersection of Fig. 5c and Fig. 6b is shown in Fig. 7a. After that, the non-intersected high roads and industrial areas have been eliminated, which is shown in Fig. 7b. Final resultant industrial areas are shown in Fig. 7c.

Prerequisites: Purulia and Hatasuria spatial databases with land use/land cover(LULC) and road informations.

System Configuration: The private cloud of IIT Kharagpur, *Meghamala* [15] has been used for this experimentation. Seven distinct VMs are used for different services. Web processing service has been assigned VM with 8 GB RAM. The web catalog service needs more space to store data or service registry and thus has been assigned 32GB persistent storage from Meghadata data service. Web map service is launched in the VMs with 4GB RAM and 2 VPUs. The orchestration engine (OE) assigns tasks using logic and accumulates all results. It has been instantiated in the VMs with 4 VCPUs and 8GB RAM. The details about the assignment of virtual machines are given in TABLE I.

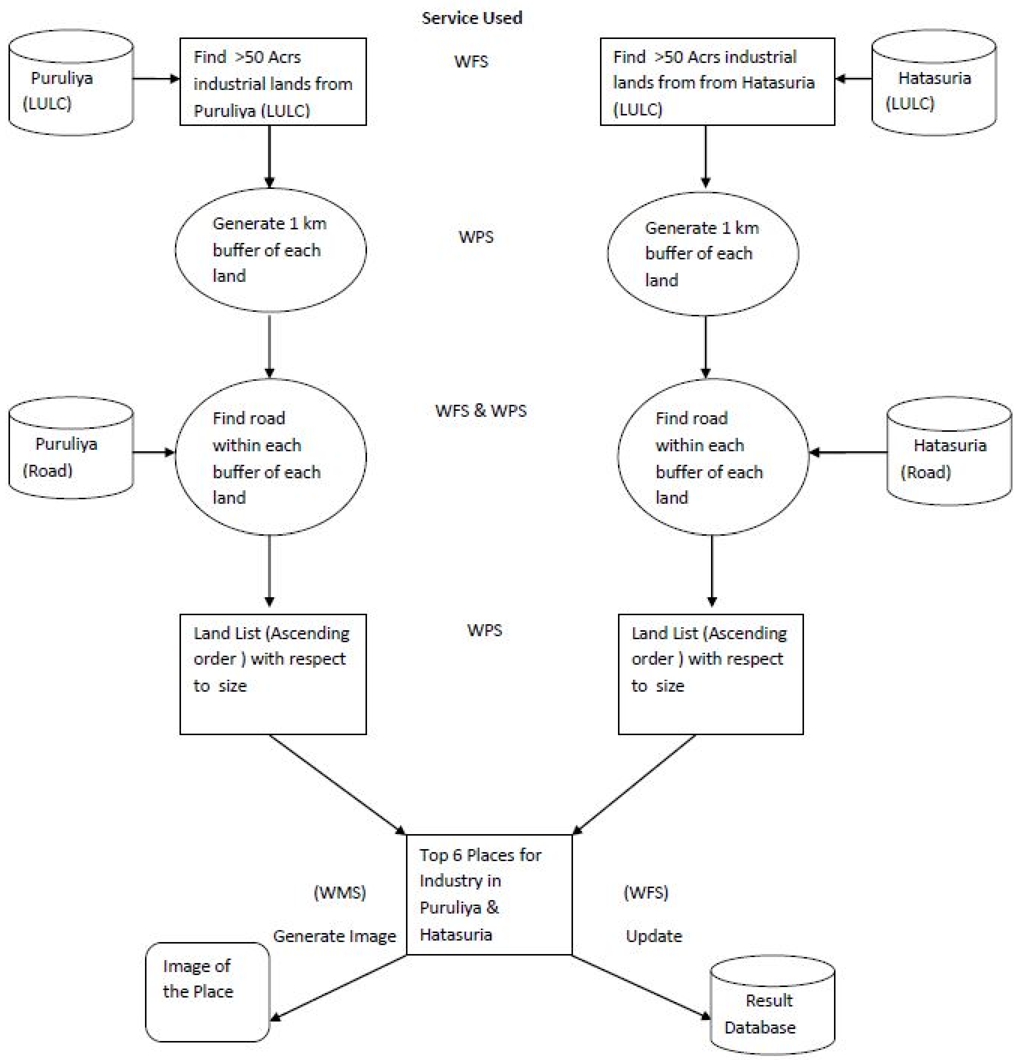

Fig. 4. Workflow of the case study

a. Purulia with areas b. Industrial areas of Purulia c. Industrial areas with 1 k.m. buffer

Fig. 5. Spatial query outputs (Study Area: Purulia, West Bengal)

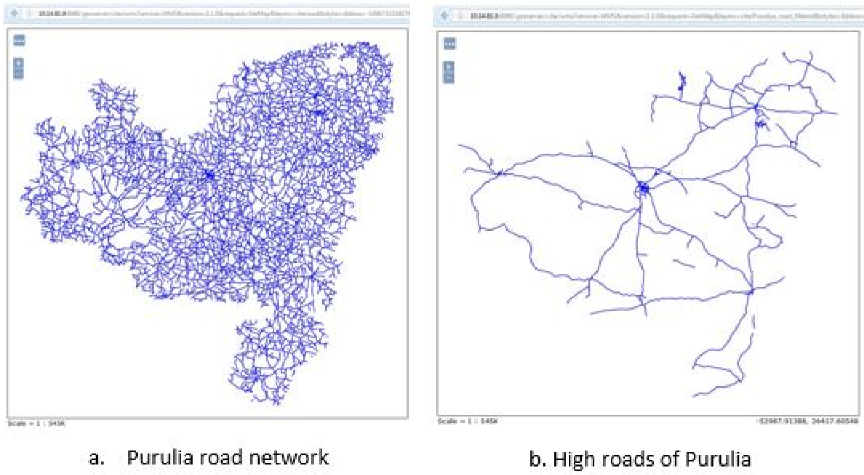

Fig. 6. Spatial query outputs for Road network (Study Area: Purulia, West Bengal)

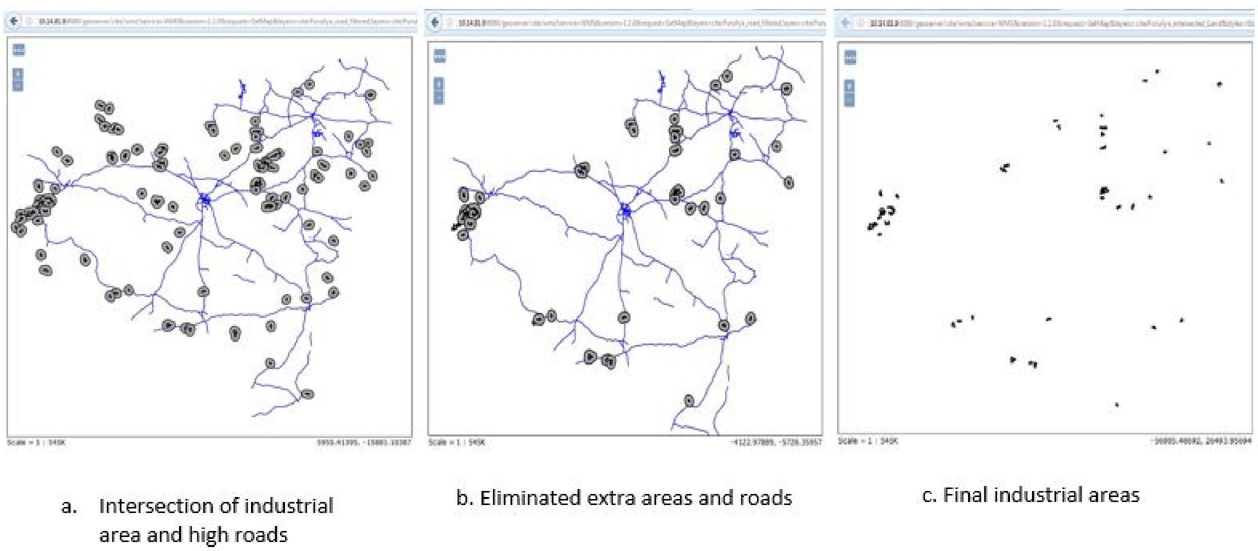

Fig. 7. Intersection of High Road and Industrial Area

TABLE I
VIRTUAL MACHINE CONFIGURATION

VM No.	Service	VM Type	VCPU Nos.	RAM(GB)	Ephemeral Storage (GB)	Persistent Storage (GB)
VM1	WFS1	IITKGP_regular	2	4	45	0
VM2	WFS2	IITKGP_regular	2	4	45	0
VM3	WPS1	IITKGP_large	4	8	45	0
VM4	WPS2	IITKGP_large	4	8	45	0
VM5	WMS	IITKGP_regular	2	4	45	0
VM6	WCS	IITKGP_regular	2	4	45	32
VM7	Orchestration Engine	IITKGP_large	4	8	45	0

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we presents a geospatial query resolution framework using an orchestration engine. The operations like filtration, buffer creation, intersection, display of data are realized which help in efficient resolution of spatial queries. The orchestration engine abstracts the user query done by feature service, processing service and map service respectively. All the available services are published with metadata in the service catalog. Sequence of services are automated by the orchestration engine getting information from catalog service.

According to the need of spatial query, synchronization of such services, executing in several virtual machines, is a challenging task. The parallel execution of some services in the cloud, may decrease the spatial query execution time. The issues related to pricing and quality of the service in the cloud can be explored in future. The cost can be optimized with the use of cost based scheduling by considering the budget of user query. A policy framework may help in controlling total task execution deadline and service response time. Performance analysis can be carried out in various public cloud platform like Amazon EC2, Microsoft Windows Azure and Google App Engine.

REFERENCES

[1] C. Yang and Q. Huang, *Spatial cloud computing: a practical approach*. CRC Press, 2013.

[2] C. Yang, M. Goodchild, Q. Huang, D. Nebert, R. Raskin, Y. Xu, M. Bambacus, and D. Fay, "Spatial cloud computing: how can the geospatial sciences use and help shape cloud computing?" *International Journal of Digital Earth*, vol. 4, no. 4, pp. 305–329, 2011.

[3] P. Yue, H. Zhou, J. Gong, and L. Hu, "Geoprocessing in cloud computing platforms–a comparative analysis," *International Journal of Digital Earth*, vol. 6, no. 4, pp. 404–425, 2013.

[4] O. Chakraborty, J. Das, A. Dasgupta, P. Mitra, and S. K. Ghosh, "A geospatial service oriented framework for disaster risk zone identification," in *International Conference on Computational Science and Its Applications (ICCSA)*. Springer, 2016, pp. 44–56.

[5] A. Dasgupta and S. Ghosh, "A framework for ubiquitous geospatial information integration on mobile device using orchestration of geoservices," *International Journal Of UbiComp (IJU)*, vol. 1, no. 3, pp. 69–88, 2010.

[6] S. S. Walia, A. Dasgupta, and S. K. Ghosh, "Geospatial orchestration framework for resolving complex user query," in *International Conference on Computational Science and Its Applications (ICCSA)*. Springer, 2011, pp. 643–651.

[7] L. Bernard and A. Wytzisk, "A web-based service architecture for distributed spatiotemporal modeling," in *Proceedings of the 5th AGILE Conference on Geographic Information Science*, 2002, pp. 25–27.

[8] P. A. Vretanos, "Web feature service implementation specification," *Open Geospatial Consortium Specification*, pp. 04–094, 2005.

[9] P. Schut and A. Whiteside, "Opengis web processing service," *OGC project document*, 2007.

[10] A. Barker and R. Buyya, "Decentralised orchestration of service-oriented scientific workflows." in *The First International Conference on Cloud Computing and Services Science (CLOSER)*, 2011, pp. 222–231.

[11] C. Peltz, "Web services orchestration and choreography," *Computer*, vol. 36, no. 10, pp. 46–52, 2003.

[12] S. Shekhar and S. Chawla, *Spatial databases: a tour*. prentice hall Upper Saddle River, NJ, 2003, vol. 2003.

[13] K. Evangelidis, K. Ntouros, S. Makridis, and C. Papatheodorou, "Geospatial services in the cloud," *Computers & Geosciences*, vol. 63, pp. 116–122, 2014.

[14] P. Yue, P. Baumann, K. Bugbee, and L. Jiang, "Towards intelligent giservices," *Earth Science Informatics*, vol. 8, no. 3, pp. 463–481, 2015.

[15] *Meghamala, IIT Kharagpur cloud service.* [Online]. Available: http://www.sit.iitkgp.ernet.in/Meghamala/