Seminar Report

GEO-CRAWLER

PREPARED BY
DEEPAK PUNJABI
15IT60R17

UNDER GUIDENCE OF
DR. S.K. GHOSH

SCHOOL OF INFORMATION TECHNOLOGY

INDIAN INSTITUTE OF TECHNOLOGY,

KHARAGPUR-721301 (INDIA)

Table of Contents

Abstract

In recent times, the increase in spatial data and services is vastly increased. To deal with this huge amount of data some kind of indexing is required. But this data is heterogeneous in nature and there are other problems as the scale of the web. We try to deal with these problems and try to implement a crawler based on *WFS* standards of *OGC* web services. Performance evaluation is an important aspect to judge the semantics used for the system.

Geospatial Information and Data

Spatial Data

The term geo comes from geography. Geography stores all the information of location and shape of the object in the spatial data. Spatial data stores the relationships between these data. It can also be easily mapped to a map. Geo-server provides various kinds of functionalities to this type of data.

Classification of Spatial Data

Different types of object under the class geometry are as below. All the major geospatial service providers and vendors provide this kind classification.

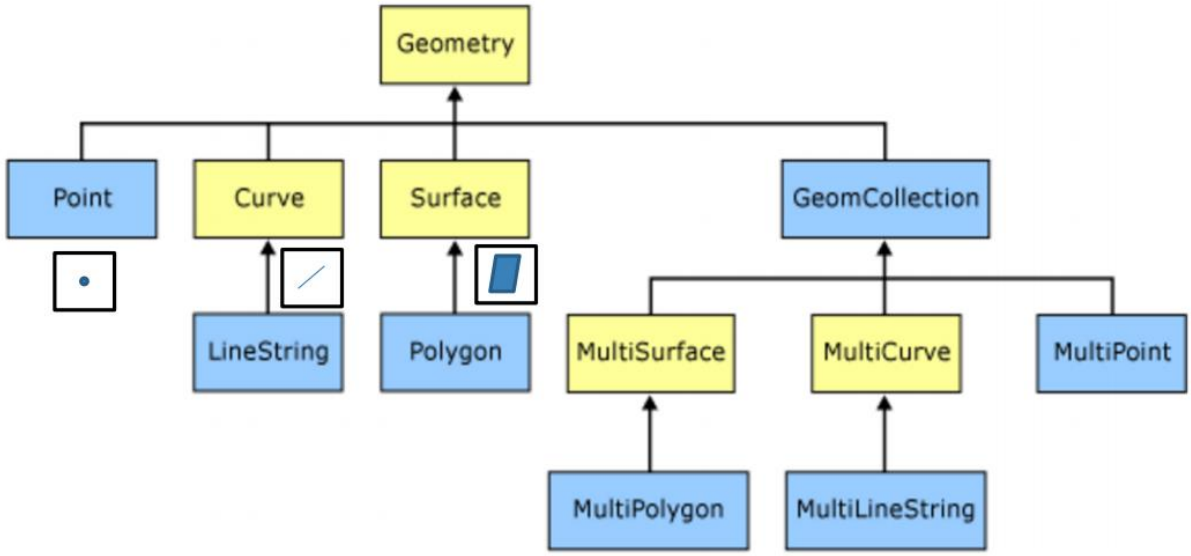

Figure 1 Spatial Object Types

The primitive four data types in spatial data are point, curve, surface and GeomCollection.

- **Point**
  Point in a map is denoted by (x, y) co-ordinates. When we see kharagpur city on a scale of India it will be seen as a point. Point can be used to denote various objects like origin, city, end point, top of the mountain etc.

- **Curve**
  Curve is used to denote collection of points. This can be a straight line or curve. For example, a road network can be represented with the help of line strings. Similarly, a river can be denoted as a curve.

- **Surface**
  A surface is representation of an area or a polygon. When kharagpur is seen in the scale of west Bengal it is seen as surface or polygon

- **GeomCollection**
  Collection of basic building blocks defining a new type of geometry can be defined with the help of GeomCollection.

NASA has a satellite called Earth Observation Satellite, which takes map images of earth and sends it to observatory. It provides 3 terabytes(TB) of data on the daily basis as per the NASA. This calculates to 90 TB data over a month and over a 1000 TB of data in a year. This data is quite huge, it is unstructured in nature and it is un-indexed. Finding relevant information out of this data is a non-trivial task. In the next section we see how the spatial data is accessed from the internet.

OGC Web Services

Open Geospatial Consortium (OGC) is the worldwide standardization body for geospatial standards. OGC provides a standardized way of accessing this geospatial data. OGC provides three kinds of web services.

- Web Map Service (WMS)
- Web Feature Service (WFS)
- Web Coverage Service (WCS)

Web Map Service

Web Map service defines a way of accessing geospatial information across all geo servers in a standard format as image. This image can be raster image or a vector image. Raster images are of type jpg, png or bmp. Vector images contains svg format extension images. It also provides a way to access metadata about the available information of the layers. This information can contain type and no of layers. Some of the well-defined operations in this layer are GetCapabilties, GetMap and DescribeLayer.

Web feature service

WFS allows direct access to features contained in the map. WFS uses SOAP based interface. For exchanging data between client and server WFS uses Geographical mark-up language which is based on XML. Some well-defined operations in WFS are query or get feature, which returns the feature stored on the server. We can add the feature in the repository by add feature. We can delete feature by delete feature. Also we can update feature stored in the repository by update feature command.

Web Coverage Service

WCS offers multi-dimensional coverage of the geo spatial data. It provides spatio-temporal context to the given geographical data. For example, it can show the flow of the river changing over the span of years. Thus we can say that WCS provides richer coverage of spatial data than WFS or WMS.

Searching of spatial data

There are previously known two popular and trivial approaches for searching spatial data.

1. **Catalogue approach**
   Service provider registers their services to the registry. The registry contains various type of registered services and their corresponding geo servers. But there are some problems with this approach. First one is that in many cases the registry is not up to date. Many times the latest services are not yet bound to any registry. One other problem is related with incorrect classification of services. This is specially an issue because user might be searching in the other part of the catalogue where it cannot find the particular service even it is there. Last kind of problem can occur because not all kind of providers registers all kind of services. Registry may be biased to some kind of services.

2. **Utilization of popular search engine**
   In an another approach we can also utilize popular search engines like Google, Yahoo, Bing or DuckDuckGo to find spatial data. But the problem with this approach is that it takes all data as general data and not spatial or other kind of unordinary data. Because of this we lose many spatial operations and features. Another problem is the popular search engines use some kind of ranking of pages which is not based on the quality of the page(QoS). For example, google uses PageRank to determine the result webpages but it is not dependent on the quality of the resultant webpage but merely a measure of from how many other webpages the result webpage is addressed.

Web Crawler

As we have seen, the above two methods fail to provide proper spatial context when searching for the geospatial data. We need something that searches the whole web searches and pre-process the data available in the internet and give optimal suggestions about available spatial service providers. A crawler is such kind of agent. It is a program that browse through the world wide web in some systematic way and creates an index of the searched data. This index can be later used for faster access for webpage or categorization of the webpage. Popular examples of crawler programs are bingbot, googlebot and polybot.

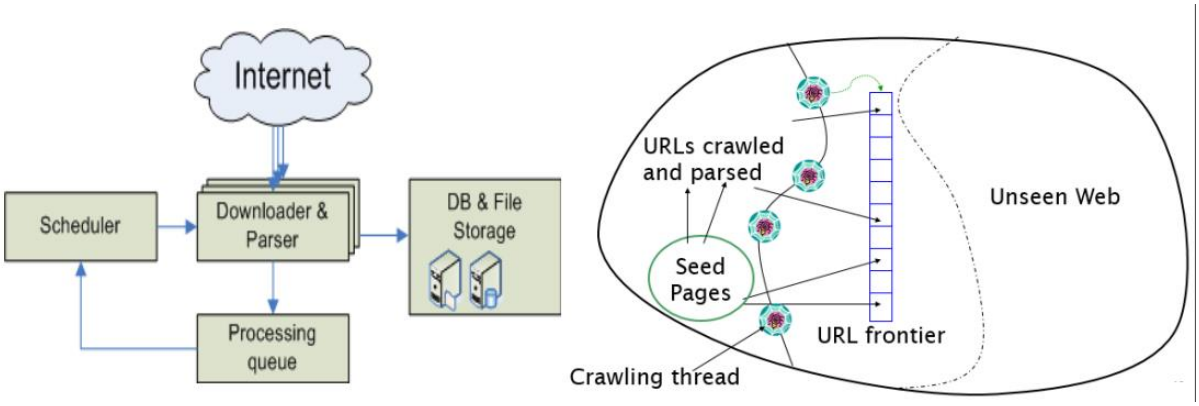

Figure 2 working of a web crawler

Challenges

There are many challenges for practically implementing a crawler in the real world. Scale of the web is so huge that the simple implementation of the theoretical crawler will be very poor. One other problem is that many pages generate, add or delete on the dynamic bases. We need to reflect this changes into the crawler. This is a crucial parameter for setting the update period or refresh rate. Another problem arises because of heterogeneity in the available data on the world wide web. How to register and store this data into the single repository depends on the implementation of the algorithm.

Types of web crawler

There are mainly two types of web crawlers

1. Universal crawler
2. Preferential crawler
    a. Focused crawler
    b. Topical crawler

Universal crawler is the crawler that treats all kind of webpages as same or given them same weight. Other kind of crawler is a preferential crawler which prefers one kind of webpage over another. One of the type of topical crawler is topical crawler. One type of topical crawler is spatial crawler to which we will look into detail.

Spatial Web Crawler: Objectives

1. Build a spatial web crawler which crawlers through geo-servers which offers WFS based OGC compliant services.
2. Build a domain specific vocabulary(ontology) for this features which can be helpful to compare found features with wanted features.
3. Perform semantic matching of found features from crawled web-pages with given ontology for filtering the correct features and storing them in the permanent repository.
4. Perform an evaluation of the given spatial web crawler using metrics and test URL seed sets.

Architecture of the Spatial Web Crawler

Our crawler contains three modules for crawling spatial features through the world wide web. Some of the definition needed for understanding the working of spatial web crawler are:

- Seed set: a set of test URLs to initialize the queue for crawling the web. The crawler starts with the URLs given in the seed URL set.
- *URLQueue*: A queue that contains set of URLs to be crawled. The crawler module takes URLs one after the another from this queue.
- Frontiers: a set of URLs from the URLQueue that are currently being crawled. This are called frontiers because they reside between the known web and the unknown web.
- WMS resolver: WMS resolver is a module that checks that if a server is WMS server or not given the URL from the URLQueue.
- Parser: parser is a module that downloads the webpage from the given URL and parse the HTML webpage looking for pre-specified tags and names. After parsing it gets a set of tags and its contents. In our implementation we parse the webpage and look for the anchor tags within it and store all the hyperlinks from the crawled webpage.
- Ontology: semantic dictionary containing all the features its type and relationship hierarchy between features.

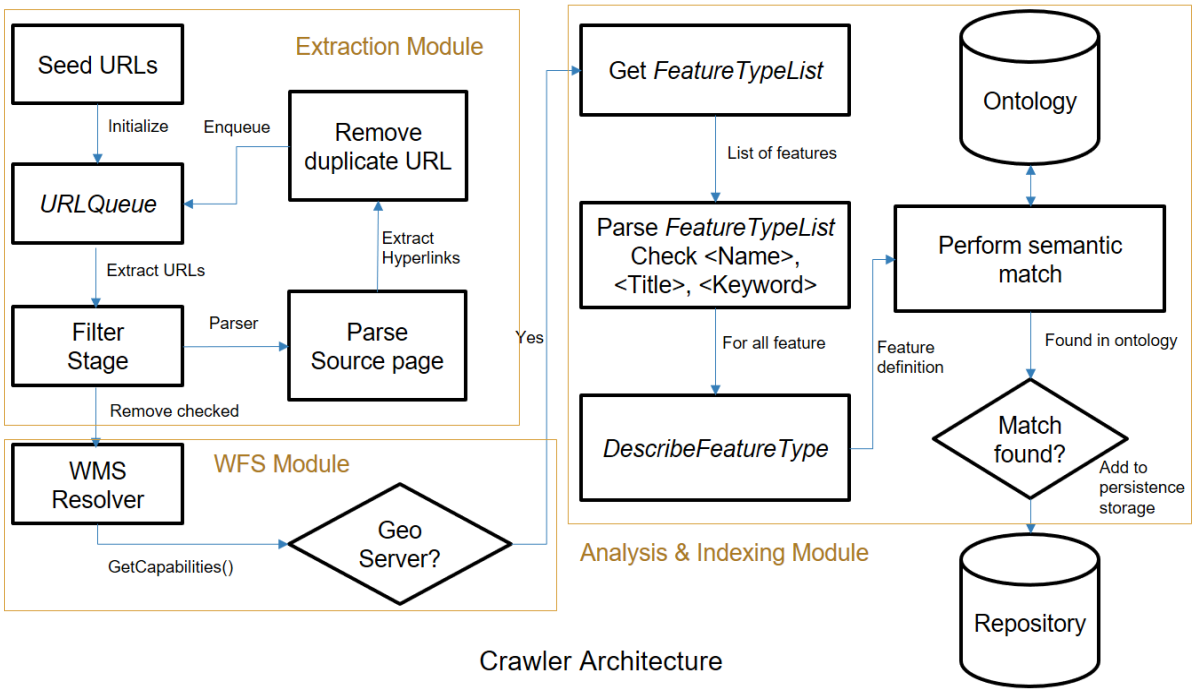

Figure 3 Architectural Diagram for Geo Crawler

Our crawler contains mainly three modules:

- Extraction module
- WFS module
- Analysis and Extraction module

Extraction module

Our algorithm starts with a set of seed URLs contained in the seed set. We initialize the URLQueue with these seed URLs. The filter stage takes URL one after the other and checks whether the given URL is already crawled or not. After filtering such URLs, they are sent to the parser. The parser downloads the webpage from the given url and parses it for finding hyperlinks contained in it. These hyperlinks again go to filter stage for finding whether the given urls are already crawled or not. After filtering these urls are added to the end of URLQueue. The filtered urls are also passed to the WFS module.

```
<?xml version="1.0"?>
- <root>
  - <FeatureType>
      <Name>prov_land</Name>
      <Title>Canadian Land</Title>
      <SRS>EPSG:42304</SRS>
      <LatLongBoundingBox maxy="83.8009" maxx="-11.9603" miny="35.8775" minx="-173.537"/>
    </FeatureType>
    -
  - <FeatureType>
      <Name>land_fn</Name>
      <Title>US Land</Title>
      <SRS>EPSG:42304</SRS>
      <LatLongBoundingBox maxy="89.8254" maxx="179.94" miny="31.8844" minx="-178.838"/>
    </FeatureType>
</root>
```

Figure 4 XML response

WFS Module

Once we have a URL for the examination, we send a GetCapabilities() request to that server. We do this by appending the request to the url.

"services?REQUEST=GetCapabilities&version=1.1.0&service=WFS"

The server replies for this request. If the reply contains <WFS_Capabilities> tag, then it offers WFS service. We parse the received response for finding the WFS_Capabilities tag. If the given server is WFS server then the given url is passed to analysis and indexing module.

Analysis and Extraction module

In this stage server response is parsed for the tag <FeatureTypeList>. <FeatureTypeList> contains list of features. This features are stored under the tag <FeatureType>. <FeatureType> tag contains set of <keyword>, <title>, <name> tags. Each of this tags are checked to see if it contains any word from the ontology. For each of such tag found, DescribeFeatureType request is appended to the url.

“?service=WFS&version=1.1.0&request=DescribeFeatureType&typename="+keyword"

Here the keyword is the name of the feature. Each of this retrieved feature is checked again the ontology, if the feature is found in the ontology then it is added to permanent storage in the repository.

Evaluation of the system

Performance Measure

The performance of the system is measured taking various seed URLs, running the algorithm against given sample data set in which correct results are already known. There are three kind of measures used for measuring the performance of the spatial web crawler.

- Precision
- Recall
- F1 measure

Precision

Precision is found by dividing the no of geo servers found by the spatial web crawler by the total no of geo servers found.

$$precision = \frac{no\ of\ relevant\ geoservers\ found}{total\ no\ of\ geoservers\ found} * 100\ \%$$

Recall

Recall is found by taking a division of no of geo servers found by the spatial web crawler by actual set of existing geo servers.

$$recall = \frac{no\ of\ relevant\ geoservers\ found}{actual\ no\ of\ existing\ geo\ servers} * 100\ \%$$

F1 measure

To normalize the values received by precision and recall parameters we generally use more robust F1 measure.

$$F1 = 2 * \frac{precision * recall}{precision + recall}$$

Final evaluation is done by taking this measure for all the feature types and averaging it over all the features.

Advantages of Spatial Web Crawler

- Allows search in web pages that are not generally searchable from the normal search engines. This is because of the spatial context awareness of the spatial web crawler.
- It provides more up-to-date results from the results. Spatial crawler is more sensitive to changes of spatial data on the web and automatically crawls through the changed features with the help of automatic update module.
- Provides improved accuracy in the search of spatial features and operations.
- Provides extra features such as bounding box of spatially crawled data and other spatial features.

Future Work

- **Priority based crawling**
  Some kind of priority can be assigned to each type of feature or operation. This can be implemented via some kind of priority queue. Usually multiple FIFO queues are implemented with some priority assigned to each queue.

- **Parallelization**
  Multi-threading and multi-core architectures can be used to help improve crawling. For example, multiple parallel threads can be created for each module with some kind of synchronization between them. This will greatly improve our performance.

- **Cloud based implementation**
  Cloud based implementation can be used to vastly scale up or scale down the functioning of the crawler. MapReduce approach can be applied where mapper node corresponds to extraction module. Reducer phase can be used to analyse and extract features out of the webpage.

- **Spatial search engine**
  After building a crawler the next step is to incorporate this crawler to some kind of spatial search engine. This search engine returns different geo spatial features upon the query of users.

- **Ranking**
  Apart from searching there is another requirement for ranking the relevant result. Chi-square test is performed on the retrieved results to improve the diversity of the results.

Conclusion

Geo-spatial data is ever growing in today's era. This type of data is hard to search from the world wide web and index it for further processing. Our algorithm suggests a way for building a WFS based geo crawler that efficiently crawls and indexes founded features into the permanent repository. This repository can then be used in many applications like search engines or data mining. This can help in many applications such as transportation & navigation, urban planning and emergency response planning.

References

I. Patil, Sonal, Shrutilipi Bhattacharjee, and Soumya K. Ghosh. "A spatial web crawler for discovering geo-servers and semantic referencing with spatial features." Distributed Computing and Internet Technology. Springer International Publishing, 2014. 68-78.

II. Li, Wenwen, Chaowei Yang, and Chongjun Yang. "An active crawler for discovering geospatial web services and their distribution pattern–a case study of OGC web map service." International Journal of Geographical Information Science 24.8 (2010): 1127-1147.

III. Jiang, Jun, Chong-jun Yang, and Ying-chao Ren. "A spatial information crawler for opengis wfs." Sixth International Conference on Advanced Optical Materials and Devices. International Society for Optics and Photonics, 2008.

IV. Marc Najork. “Web crawler architecture.” Microsoft Research.

V. Ahlers, Dirk, and Susanne Boll. "Location-based Web search." The Geospatial Web. Springer London, 2009. 55-66.

VI. Li, W., et al. "Semantic-based web service discovery and chaining for building an Arctic spatial data infrastructure." Computers & Geosciences 37.11 (2011): 1752-1762.