A Spatial Information Crawler for OpenGIS WFS

JIANG Jun*, YANG Chong-jun, REN Ying-chao
The State Key Laboratory of Remote Sensing Information Sciences, IRSA, CAS, Beijing, 100101

ABSTRACT

The growth of the internet makes it non-trivial to search for the accuracy information efficiently. Topical crawler, which is aiming at a certain area, attracts more and more intention now because it can help people to find out what they need. Furthermore, with the OpenGIS WFS (Web Feature Service) Specification developed by OGC (Open GIS Consortium), much more geospatial data providers adopt this protocol to publish their data on the internet. In this case, a crawler which is aiming at the WFS servers can help people to find the geospatial data from WFS servers. In this paper, we propose a prototype system of a WFS crawler based on the OpenGIS WFS Specification. The crawler architecture, working principles, and detailed function of each component are introduced. This crawler is capable of discovering WFS servers dynamically, saving and updating the service contents of the servers. The data collect by the crawler can be supported to a geospatial data search engine as its data source.

Keywords: web crawler, search engine, WFS, OpenGIS, geospatial information

1 Introduction

The web information is growing so rapidly that it is difficult to retrieval the accurate information from the web in a timely manner. In order to solve the problem, web search engine is widely used by the people who surf the web. [1] However, there is too much information, and people are not satisfied with the results from the integrated web search engines. When people focus on a certain area, they need the engines to provide all the results specifically on this topic.[2] With this challenge, topical web search engines become hotspot in current research, furthermore, topical crawler, an important part of the web search engines attracts the researcher's attention.

1.1 Topical crawler

Topical crawler is a crawler on the web aiming at web pages which are on a certain topic. The difference between a general crawler and a topical crawler is that the topical crawler has a special module to parse the pages contents in order to exclude the web pages which are not interrelated with the topic.[3]

1.2 Geospatial information on the Internet

Geospatial information is the information related with the geographic position. With the progress of the remote sensing, GPS (Global Position System) and GIS, geospatial information plays an important role in everyday life. The web search engines which focus on the space information provide a connection between the geospatial information servers and the users, thus advance the share of the geospatial information. [4]

There are two methods in which the geospatial information servers provide the information on web: the static and the dynamic. [5] The static method is to publish the geospatial information in static web pages as texts or static images. The dynamic method is that the servers response to the queries from the users, providing the results in a dynamic response method. [6]At present the search engines on the static web pages have made great progress such as Google,Baidu,etc. On the other hand, the systems aiming at the web spatial information servers which responses to the dynamic queries are not as familiar and do not evolve very well. As more and more spatial information servers adopting the OpenGIS Web Feature Service (WFS) Specification, a crawler which can search the spatial information servers based on the OpenGIS WFS is needed.

* e-mail: jiangjun224@gmail.com

Geoinformatics 2008 and Joint Conference on GIS and Built Environment: Geo-Simulation and Virtual GIS Environments, Lin Liu, Xia Li, Kai Liu, Xinchang Zhang, Aijun Chen, Eds., Proc. of SPIE Vol. 7143 71432C · © 2008 SPIE · CCC code: 0277-786X/08/$18 · doi: 10.1117/12.812614

2 Related works

At present, the interest of research of dynamic space information search focus on the Geospatial Data Clearinghouse and the geography Network. [7]

2.1 Geospatial Data Clearinghouses

Geospatial Data Clearinghouse is a research plan to share the geospatial data between the provinces on the internet carried out by American FGDC (Federal Geographic Data Committee). Geospatial Data Clearinghouse is a distributed network which connects geospatial data providers, governors and consumers. Based on its support, geospatial data consumers can access the data can be used now and find out what they need, further more, they can evaluate these data and fetch them in the minimum cost.

Now, American Geospatial data interchange center collects more than 250 geospatial data servers and hundreds of databases. There are 6 data gateways to provide the access to the data queries, data set can be fetched via a user interface or metadata query.

Geospatial Data Clearinghouse helps the users to find out the geospatial data providers as the connection between the users and providers. However, it can not help users to fetch the real-time data because it doesn't cover the data transport standards and specifications. Users must connect the providers in other methods.

2.2 Geography Network

Geography Network is presented by ESRI Company. It is a global network of geospatial data users and providers. The network provides geography data by internet viewer and GIS desktop systems. These data include some unprocessed data, maps and some related services. Geography Network provides the index of geography data and service. Users can visit it via general viewer or some GIS desktop systems like ArcInfo, ArcView GIS and ArcExplorer. Geography Network makes a great progress on the online publish of the geospatial data and service, and it also facilitates the flexible sharing of the data. However, only the ESRI Company's systems can be included and the unregistered data can not be fetched.

Since the OGC (Open Geospatial Consortium) established the OpenGIS Web Feature Service Implementation Specification, more and more geospatial data providers adopt this specification to publish their data on the internet. This specification supports the data query and the data transportation in standards, so the users can get the real time processed data from the server on the internet. The search engines aim at the WFS servers can help users to find the geospatial data from WFS servers, which are seldom. Therefore, this paper presents ideas about the search of the WFS servers.

3 The WFS Specification

A WFS (Web Feature Service) server publishes feature-level geospatial data to the web, *i.e.*, instead of returning an image, as MapServer has traditionally done, the client now obtains fine-grained information about specific geospatial features of the underlying data, at both the geometry AND attributes levels. As with other OGC specifications, this interface uses XML over HTTP as its delivery mechanism, and, more precisely, GML (Geography Markup Language), a subset of XML. The WFS specification defines interfaces for data access and manipulation operations on geographic features, using Http as the distributed computing platform. Via these interfaces, a web user or service can combine using and managing geospatial data—the feature information behind a map image—from different sources. [8]

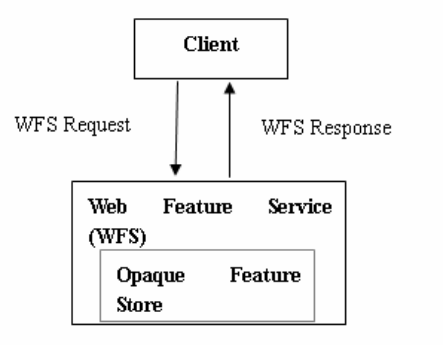

Figure 1: WFS Specification

The main interfaces provided by WFS are: GetCapabilities, GetFeatureType and GetFeature.

Table 1: the main operations of the WFS server

The operations	the introduction
GetCapabilities	Return the service capabilities such as the version, the name and the content
GetFeature	Return the features offered by the server
DescribeFeatureType	Return the feature type

3.1 GetCapabilities Operation

The GetCapabilities operation returns the service capabilities such as the version, the name and the content. It also returns which feature types the server can provide and which operations are supported on each of them. It enables users to specify which feature properties to fetch via the GetFeature interface, which returns GML files containing the feature instances the users need. [9]

As the WFS provide the GetCapabilities interface, the crawler judges whether the server for WFS relied on its response to the GetCapbilities query. The principle of the crawler is to crawl on the web sites and analyze the responses.

3.2 DescribeFeatureType Operation

The DescribeFeatureType operation returns the XSD schema information for a set of FeatureTypes. The schema describes the property names and types associated with the FeatureType. A FeatureType is a *single* concrete type of GML feature (as defined by an Application Schema .XSD). It is usually named with a "prefix:name" tag to uniquely identify it relative to other FeatureType names. A FeatureType is often (but not always) synonymous with a dataset or layer. A dataset may contain multiple FeatureTypes - for example, the "City" dataset may contain "ny:Road" and "ny:Building" FeatureTypes. Make a DescribeFeatureType request when users want to know more detailed information about the schema for a FeatureType. The response will contain the FeatureType's property names, types, and restrictions.

3.3 GetFeature Operation

The GetFeature operation allows retrieval of features from a Web Feature Service. An XML document containing the result set is returned to the client. A GetFeature request can be issued through both GTTP GET/KVP and GTTP POST/XML encodings. The GetFeature element has an optional *maxFeatures* attribute that restricts the number of features to be retrieved. Without this attribute, all available features are retrieved. It is highly recommended to use *maxFeatures*. Otherwise, responses can become overly long, and some clients may not be able to properly parse the response.

4 System Anatomies

4.1 The structure of the crawler

4.1.1 The crawl module of the crawler

There are two primary modules in the crawler. The first module crawls on the internet. With a start URL, the crawler retrieves the web page, analyzes the content and finds out all the internet links in this page. Then the crawler stores this URL and all the links into a temporary URL database. After analyzing the whole start page, the crawler gets another URL from the temporary database and repeats the work above. Because the useful web pages on the internet are never unlinked from other web sites, the crawler can always crawl on the internet with the links. [10]

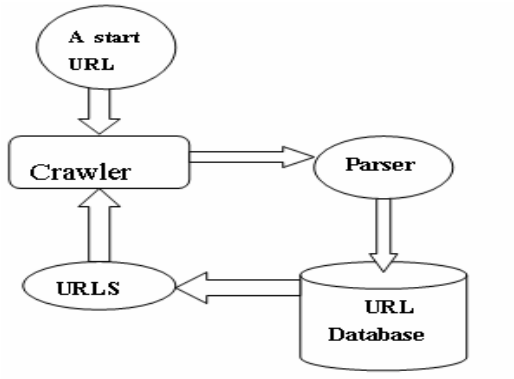

Figure 2: the crawl part of the crawler

In order to improve the efficiency, there are a number of threads to operate the crawler programs. These threads work at the same time continuously. However, these may not be enough for big systems. A large number of high-powered computers work together to crawl more web pages.

4.1.2 The Parse module of the crawler

The other module of the crawler determines whether the web site is a WFS server. The crawler gets a web site from the temporary database, and assumes the web site is a WFS server and can response to the GetCapabilities query. Then, the crawler sends a GetCapabilities request to the web site. If the web site does not return any information in a specified period, the crawler turns to get another URL from the database. Otherwise the web site response to the query, the crawler analyzes the answer and judges whether it is a WFS server. On condition that the answer is positive, the crawler takes further operation to get the spatial extent of the service. Finally, the crawler saves the URL and the spatial extent into the database.

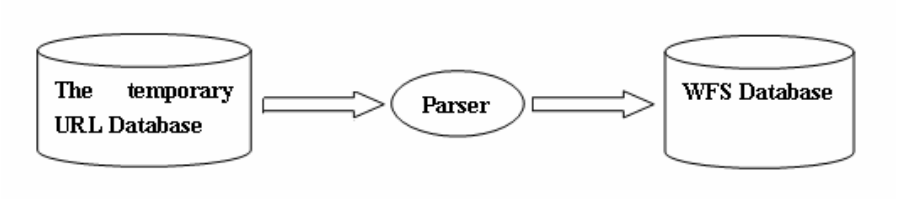

Figure 3: The analyze part of the Crawler

4.1.3 Other details of the crawler

The two modules of the crawler need to work at the same time. As a result, the crawler starts two thread groups. Thread group I takes charge of crawling, and thread group II checks the WFS servers. Thread group I has precedence over thread group II.

With the structure introduced above, the crawler can crawl on the web pages. However, we need to consider more details to advance the system for real applications. Crawling on the web pages requires both efficiency and reliability. It is not an easy task to work with millions of web servers in dynamic. [11]

The crawler is realized by a number of threads in a computer, and these threads adopt critical sect to work synchronously. To solve the problem that the remote servers stop working abruptly, the crawler threads set back-up address database to store the web pages that are not linked successfully temporarily, then these web pages will be distributed to the threads to connect again in a proper time.

For the crawler can crawl more web pages, the system needs to consider the fetch and the analyze of some dynamic pages like ASP,JSP and PHP pages as well as the static html web pages. In this condition, the crawler requests these dynamic web pages and pick up the links on these pages.

On the other hand, the number of the web sites on the internet is so huge that it is a heavy burden for the crawler to check all the URLs in the temporary database to judge the WFS servers. Because the WFS server provides dynamic services, the crawler can eliminate all the static web pages in order to reduce the workload. Most static web pages' URLs end with the words ".html" or ".htm", so the crawler can exclude some static web sites relied on the URLs. Furthermore, most dynamic web sites are linked with the sign "?" by other web pages and the links contain some additional query words, so the crawler needs to get rid of the query words to extract the real URLs of the web sites.

The crawler must start, pause and stop immediately under the control of the main thread. In order to conserver the fieldwork of the crawl, when the main thread orders to stop, a separate thread will be waken to store the unfinished web page to a special database. Then the crawl work start again with the unfinished web page which is stored, as a result, the crawl can continue without difference.

Redundancy is another problem of the web search. Some links are IP addresses of the web pages, while others are domain names. For example, the domain name "www.pku.edu.cn" and the IP "162.105.129.12" point to the same web page. [12] All the DNS (Domain Name System) servers provide the parse from the domain names to the IP addresses, but not all of them provide the reverse parse from IP addresses to domain names. In addition, not all the IP addresses correspond with domain names, so the crawler change every domain name link to IP address link in order to avoid the repeat crawl. However, the queries to the DNS server really cost excessive time, so the crawler set a DNS cache to store some DNS records. If the domain name which needs to be parsed is in the cache, the crawler can get the IP address in time, without querying to the DNS server again. When the scale of the web research is not big, the DNS cache can be set in the memory. The practice proves that this method is quite effective.[13]

4.2 How to analyze the WFS pages

When the crawler determines whether a URL is WFS page, at first, it need construct a query to the server. Based on GetCapbilities operation of the WFS Specification, the query needs contain the version, the operation name, and the service. For example:

Using HTTP GET:

http://www2.dmsolutions.ca/cgi-bin/mswfs_gmap?version=1.0.0&request=getcapabilities&service=wfs

Using HTTP POST:

```
<?xml version="1.0"?>
<wfs:GetCapabilities
        Service="WFS"
        Version="1.0.0"
        Xmlns:wfs="http://www.opengis.net/wfs"
        Xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
        Xsi:schemaLocation=http://www.opengis.net/wfs
http://schemas.opengis.net/wfs/1.0.0/wfs-basic.xsd
/>
```

When the crawler gets the response, it analyzes the XML file. If the XML file contains the key words like "WFS_Capabilities", the crawler can decide that the web page is WFS server. Then, the crawler needs to parse the other contents of the XML file. In order that the WFS database can provide more information about the WFS servers than just a URL, the crawler has to record the bounds and some other contents of the service.

Table 2: the data structure in the database

Field name	description	type	NULL
URL	The urls of the services	nvarchar	no
abstract	The description of the service	nvarchar	yes
minX	The min x of the bound	double	yes
minY	The min Y of the bound	double	yes
maxX	The max X of the bound	double	yes
maxY	The max Y of the bound	double	yes
updateTime	The update time of the URL	DateTime	no

The data table has the field "update Time" to record the update time of the URLs. If the update time is so close that does not exceed a certain period, the crawler doesn't request the URL again. As a result, the efficiency can be advanced and the case that the crawl work is trapped in a dead cycle because of the repeat search can be avoided. The update period of a crawler is very import. If the period is too short, the burden of system is heavy, and if the update period is long, the some mistake links can not be found in time. Therefore, how to define a proper update period should be considered.

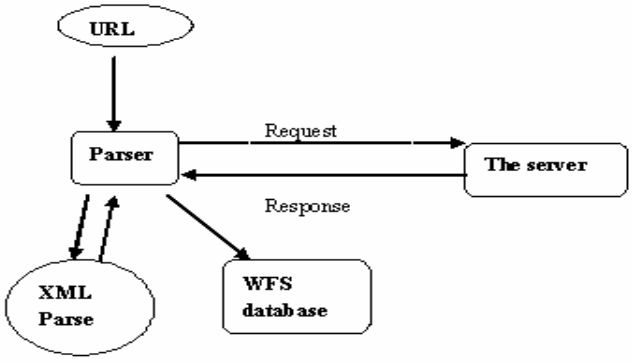

Figure 4:The structure of the analyze part of the crawler

4.3 The analyses of the XML file

XML is a markup language much like HTML, and was created to structure, store, and transport information. It is amazing to see how quickly the XML standard has developed and how quickly a large number of software vendors have adopted the standard. XML is now as important for the Web as HTML was to the foundation of the Web. XML is everywhere. It is the most common tool for data transmissions between all sorts of applications, and becomes more and more popular in the area of storing and describing information.

XML documents form a tree structure and it has tags which are flexible. Sample response from the WFS server in the GetCapabilities operation is as follows:

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

At present, there are some XML parsers that can be used to read and manipulate XML. The parser reads XML into memory and converts it into an XML DOM object that can be accesses with JavaScript. In our crawler, it is not necessary to analyze all the data that carried by the XML file, so we do not need to use a professional parser to analyze the XML file. What we need is to find out the bound of the service and store it in the WFS database. As a result, we just find the key words of the bound like “minx,miny,maxx,maxy”, then get the data behind the key words. Because in the XML file, there are a number of entities that have bound box, a cycle of finding out the max bound must be contained in the program.

5 The implementation of the crawler

The implementation of the crawler is carried out on the windows XP operating system, using vc6.0 as the develop tool and SQL Server 2000 as the database.

The steps of the crawler’s work are:

1. The user appoints a start URL to the crawler or chooses a URL offered.
2. the crawler fetch the web page and get all the links on the page
3. the crawler stores all the links to the temporary URL database
4. the crawl module of the crawler continues crawling the web page by getting a URL from the temporary URL database
5. the parser module of the crawler get a URL from the temporary database and make a GetCapabilities query to it
6. if the parser module get a XML file back from the web page, it analyze the XML file to judge the WFS server, or else the parser module goes back to Step 5
7. if the web page is a WFS server, the parser module store the URL and some other contents to the database, or else the parser module goes back to Step 5

When the crawler is working, the number of the URLs stored by the crawl module to the temporary database is huge because the crawl module crawls on the web fast. However, the WFS servers that are stored to the database do not increase so rapidly.

The time to parse the XML file is not very long, but the WFS servers distribute on the web so that it is not as easy to find them as the general web pages.

The number of the WFS servers recorded in the database is not very huge. When users need to find services bounded in a certain area, it is fast to search the database and fetch the proper URL. However, for general search engines, the data fetched by the crawler from web pages is so much that a simple database can not meet the need. As a result, big search engines usually save the data on the disk in some formats to process them for the service. The storage formats include the structure and the index of the data. The structure design should consider that the data needs to be saved for a long time and it must be easily processed. Further more, because the disks have their own life span, the structure of data also needs to advantage the resume of the data when it is attainted. Without the index of the data, it takes a long time and much computer resource to find the record users need. Different search engines adopt different indexes, and the quality of the index determines the response time and the accuracy of a search engine. Therefore, every successful search engine must have an effective index of the data.

For general big scale search engines, another important work is to evaluate the web pages. When these search engines response to users' queries, the web pages' value determine the pages order in the result. The most well known web page rank technique is Google's PageRank. PageRank is based on the liberty and the numerous of the web links. In these plenty of links, Google picks up millions of them to analyze, and draws a huge web map. According as this web map, PageRank can compute the rank of a web page very fast. This complex but auto rank method excludes the influences on the results by people. As a result, most people prefer the search results offered by Google, and this is one of the reasons that why Google grows so rapidly.

6 Conclusions

6.1 The conclusion of the crawler

This paper analyzes the publish manner and characteristics of the geospatial data provided on the internet at present. With the OpenGIS WFS Specification is adopted by more and more data provider, this paper presents the design and the implementation of a crawler based on the OpenGIS WFS Specification.

The research significance of this paper is that with more and more geospatial data provided on the internet, people are fronted with the problem to find the useful geospatial data effectively. As the general web search engines can help find out the useful web pages, the geospatial data crawler can help find the WFS servers and their service contents. [14]Therefore, the crawler constructs a bridge between the WFS servers and the geospatial data users. With the OpenGIS WFS is becoming more important, the crawler aiming at it is also more significant.

6.2 Further work

Many WFS servers are provided to general users so that much further work can be carried out based on the crawler's work described in this paper. A spatial information search engine can have the WFS servers found by the crawler as its data source. When the search engine is designed, much more details should be considered. As the number of the WFS server records is increasing rapidly, how to index the web pages is becoming crucial. Unlike Google aiming at the general web pages without discrimination, the spatial WFS search engine concentrates on the WFS on the internet so that much more spatial information will be well utilized and general users can have access to more and more spatial information.

References

[1]The Anatomy of a Large-Scale Hypertextual Web Search Engine, Sergey Brin and Lawrence Page

[2] Design and Implementation of a Web Page—gathering Tool, PAN Chun-hun． CHANG Min WU Gang—shah, Application Research of Computers,2006

[3] Design and Implementation of Topic—specific Personal Real—time Search Engine, Liu Jieqing W u Jinghui, New Technology of Library and Information Service,2006.05

[4] Design and Implementation of Spider on Web—based Fu l l—text Search Engine, XU YUANCHAO LIU JIANGHUA LIU LIZHEN GUAN YONG

[5] OpenGIS WMS Based Prototype of Spatial Information Search Engine, BA I Yuqi, YAN G Chongjun, L IU Donglin, ZHU Huaji, LU Yahui, RU I Xiaoping, Journal of Image and Graphics

[6] Prototype of Spatial Information Search Engine, BAI YUQI, 2003 IEEE International Geoscience and Remote Sensing Symposium (IGARSS'03)

[7] Spatial Search Engine - Enabling the Intelligent Geographic Information Retrieval, BAI YUQI, Proceedings of ISPRS Commission II Symposium,Xi'an, 2002.8

[8] OGC WFS Specification: 04-094_Web_Feature_Service_Implementation_Specification

[9] Feature-level geospatial data sharing: a case study for disaster precaution space, Kuo-chih Hung, Feng-Tyan Lin

[10] Research on Spatial Information Search Engine, BAI YUQI, Journal of China University of Mining & Technology,2004,1

[11] Survey on topic—focused Web crawler, LIU Jin·hongLU Yu·lian, Application Research of Computers, 2007,10

[12] Search Engine: Principle, Technology and Systems, Li Xiaoming, Yan Hongfei and Wang Jimin , Science Press

[13] Summary of Web Data Mining and Personalized Search Engine, ZOU Fang—hong

[14] GIS Inter0perability and OGC Specifications, LI Xin—tong', HE Jian—bang, GEOMATICS W ORLD,2003,10