International Journal of
Geo-Information

Article

GeoWeb Crawler: An Extensible and Scalable Web Crawling Framework for Discovering Geospatial Web Resources

Chih-Yuan Huang [1,*] and Hao Chang [2]

[1] Center for Space and Remote Sensing Research, National Central University, Taoyuan 320, Taiwan
[2] Department of Civil Engineering, National Central University, Taoyuan 320, Taiwan; 103322088@cc.ncu.edu.tw
* Correspondence: cyhuang@csrsr.ncu.edu.tw; Tel.: +886-3-422-7151 (ext. 57692)

Academic Editor: Wolfgang Kainz
Received: 8 June 2016; Accepted: 29 July 2016; Published: 5 August 2016

Abstract: With the advance of the World-Wide Web (WWW) technology, people can easily share content on the Web, including geospatial data and web services. Thus, the "big geospatial data management" issues start attracting attention. Among the big geospatial data issues, this research focuses on discovering distributed geospatial resources. As resources are scattered on the WWW, users cannot find resources of their interests efficiently. While the WWW has Web search engines addressing web resource discovery issues, we envision that the geospatial Web (i.e., GeoWeb) also requires GeoWeb search engines. To realize a GeoWeb search engine, one of the first steps is to proactively discover GeoWeb resources on the WWW. Hence, in this study, we propose the GeoWeb Crawler, an extensible Web crawling framework that can find various types of GeoWeb resources, such as Open Geospatial Consortium (OGC) web services, Keyhole Markup Language (KML) and Environmental Systems Research Institute, Inc (ESRI) Shapefiles. In addition, we apply the distributed computing concept to promote the performance of the GeoWeb Crawler. The result shows that for 10 targeted resources types, the GeoWeb Crawler discovered 7351 geospatial services and 194,003 datasets. As a result, the proposed GeoWeb Crawler framework is proven to be extensible and scalable to provide a comprehensive index of GeoWeb.

Keywords: Geospatial Web; resource discovery; Web crawler; Open Geospatial Consortium

1. Introduction

1.1. Big Geospatial Data

Geospatial data are used to help decision making, design, and statistic in various fields, such as administration, financial analyses, and scientific researches [1–4]. With the advance of the World-Wide Web (WWW), the Web 2.0 represents a concept that allows everyone to easily set up websites or post content on the Web [5]. Users can share their data or services on different Web 2.0 platforms, such as Facebook, Wikipedia, and YouTube. Among all the resources (including data and services) available online, the resources that contain geospatial information constitute the "GeoWeb" [6]. The main objective of the GeoWeb is to provide access to geospatial data or services through the worldwide coverage of the WWW. For example, one of the most famous GeoWeb resources is the web mapping service, such as Google Maps [7]. According to [6], GeoWeb is based on a framework of open standards and standards-based technologies, such as geospatial services and Spatial Data Infrastructure (SDI) [8]. Users can then connect to these resources to retrieve geospatial data/information for further analyses.

While more and more data on the Web are generated faster than at any time in the past, the concept of "big data" starts attracting attentions. Laney [9] presented the 3V characteristics of the big data, which are volume, velocity, and variety. Specifically, data on the Web are generated in high rate (velocity), require large storages (volume), and have various types (variety), such as text, image, and video. Data in the GeoWeb also fit into the 3V characteristic of big data [10]. For instance, the satellite images would cause the storage issue since each image size may be up to several gigabytes. For the velocity, due to the advance of sensor and communication technology, the number of sensors is rapidly increasing, and sensors are generating observations at high frequency. In addition, the sensor observations in GeoWeb [11] are produced by a large number of sensor nodes that monitor various kinds of phenomenon, such as temperature, humidity, and air pressure, and have a large variety regarding data formats, web protocols, and semantic frameworks.

1.2. Problems in Geospatial Resources Discovery

While the big data provide opportunities to discover phenomena that were previously undiscoverable, various big data management issues should be addressed [10]. Among the issues, this study mainly focuses on addressing the issue of GeoWeb resource discovery. Currently, if scientists want to search geospatial data on the WWW, they usually start with finding a data portal or an SDI. The data portals are the web sites/services that data providers host, such as the NASA's data portal (https://data.nasa.gov/) and National Climatic Data Center (http://www.ncdc.noaa.gov/). SDIs are mainly the catalog and registry services provide entries of data services. While providers can manage services on their own, users can find the service entries from SDIs and access the services directly. The INSPIRE (http://inspire-geoportal.ec.europa.eu/), Data.gov (http://www.data.gov/), and GEOSS (http://www.geoportal.org/web/guest/geo_home_stp) are examples of SDIs.

However, a critical issue of the existing approaches is that a user may not know where to find the data portals or SDIs that provide the data or services the user needs. Besides, while data in portals are hosted by different organizations and data in SDIs are registered by different providers, no single data portal or SDI can provide complete data index on the GeoWeb. In addition, the heterogeneity problem exists in user interfaces and data between each portal or SDI. Users need to spend more time on learning the procedures to access data and understanding the data. In terms of SDI, the complex registry and lack of incentive would reduce the willingness for providers to register their data. In general, these problems (the details will be further discussed in Section 2.1) would cause GeoWeb resources discovery inefficient.

Furthermore, resources outside geospatial data portals/SDIs are important as well. Based on [12], the GeoWeb resources can be modeled as a long tail (Figure 1). While most geospatial data portals and SDIs are managed by a small number of government or research institutes, these portals and SDIs provide a substantial amount of geospatial data. These data are called as the "head" in the long tail and usually accessible to users. On the other hand, there are small data portals and GeoWeb resources hosted by a large number of individual scientists or small institutes, which comprise the "tail". According to the long tail theory [13], the amount of data in the tail equals to the amount in the head. However, the resources in the tail are not indexed by the data portals or SDIs in the head, and usually are undiscoverable to most users. Besides, one of the important features of GIS (Geographic Information System) is overlaying interdisciplinary layers and discovering the relations between them. Especially for data scientists, they may not always know where to find the geospatial data they need [14]. In this case, it is important to have a repository to organize various types of geospatial data. Therefore, in order to provide users a complete index of GeoWeb and a one-stop entry to find every GeoWeb resource, we need a solution that can automatically discover every geospatial resource shared online.

In order to make a comprehensive discovery on the WWW, one of the most effective solutions is the web search engine, such as Google, Bing, and Yahoo!. These search engines apply the web crawler approach to find every web pages by traversing through every hyperlink [15,16]. However, search for geospatial resources is not easy since most of the search engines are designed to search plain-text web pages. Users still need to identify geospatial resources from web pages by themselves. In addition, GeoWeb has various resource types, such as Open Geospatial Consortium (OGC) web services, geospatial data like GeoTiff, GeoJSON, and Shapefile, and catalogue services. There is no general solution to identify every type of resource. Based on these reasons, GeoWeb needs a specific search engine with an extensible crawling mechanism to identify different types of GeoWeb resource.

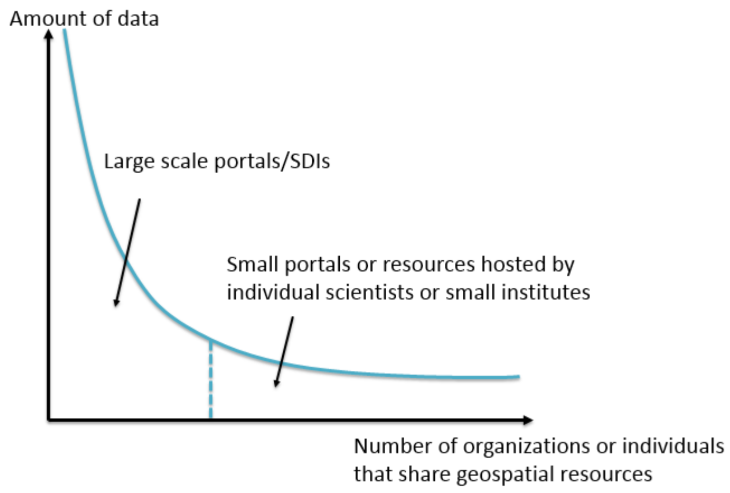


Figure 1. The GeoWeb Long Tail.

In addition, following open standards is the key to easily identify GeoWeb resources. Since open standards regulate the client-server communication protocols and data models/formats, GeoWeb resources can be interoperable as long as client and server follow open standards, such as standards in OGC and International Organization for Standardization (ISO). Otherwise, in terms of the resources that are published following proprietary protocols. Although these proprietary resources could also be identified, there is an additional cost to customize specific connectors.

1.3. Research Objective

For building a GeoWeb search engine, the first step is to collect a comprehensive index of GeoWeb resources. Hence, the main focus of this study is to identify and index every geospatial resource on the WWW. To be specific, we propose a large-scale GeoWeb crawling framework, the GeoWeb Crawler, to proactively discover GeoWeb resources. Although web crawlers have been wildly used in web search engines to discover web pages, the GeoWeb Crawler needs to be extensible to identify various types of geospatial resources. Therefore, one of the keys in this study is to define the identification rules for different GeoWeb resource types and apply these rules in a crawling framework. In this case, the GeoWeb Crawler can be easily extended to find other resource types. In addition, as web crawling is a time-consuming process, in order to discover geospatial resources efficiently, we applied the distributed computing concept [17] in the GeoWeb Crawler to execute the crawling process in parallel. Overall, this research proposed an extensible and scalable GeoWeb crawling framework that has the potential to proactively identify every geospatial resource shared on the Web. We believe that the GeoWeb Crawler could be the foundation for a GeoWeb search engine.

The remainder of this paper is organized as follows. Section 2 discusses existing approaches for GeoWeb resources discovery. Section 3 defines the GeoWeb resources distribution and research scope. In Section 4, we present the details of the GeoWeb Crawler, including extracting Uniform Resource Locators (URLs) from plain-text web pages, identification of different types of resource, distributed computing workflow, and filtering of redundant URLs. Section 5 explains the crawling results and evaluations from different perspectives. In Section 6, we demonstrate the prototype of a GeoWeb search engine, called GeoHub, based on the resources discovered by the GeoWeb Crawler. Finally, the conclusions and future work of this study are presented in Section 7.

2. Related Work

2.1. Existing Approaches for GeoWeb Resource Discovery

As aforementioned, we categorize the existing approaches for finding GeoWeb resources into the data portal/service and catalog approaches. These two approaches can be further separated into open-standard solutions and non-open-standard solutions, as shown in Table 1. Web services such as OGC web service and Open Source Geospatial Foundation (OSGeo) Tile Map Service are the examples for the data portal/service following open standards. The geospatial resources are published according to the specifications. On the other hand, most data portals such as the NASA's data portal and National Climatic Data Center are not based on any open standard. The data models and communication protocols of these non-open-standard portals/services are different. For the catalogs, they are the entrances to find resources. OGC Catalogue Service for the Web (CSW) is an open standard for regulating the catalog service interface while SDIs such as Data.gov and GEOSS are developed independently. Here we discuss the problems of these four categories.

Table 1. Categories of existing GeoWeb resource discovery approaches.

	Open-Standard	**Non-Open-Standard**
Data portal/service	Web services, e.g., OGC web services and OSGeo Tile Map Service (TMS)	Data portals, e.g., National Aeronautics and Space Administration (NASA)'s data portal, National Climatic Data Center, and National Oceanic and Atmospheric Administration (NOAA)'s tides and currents.
Catalog	OGC Catalogue Service for the Web (CSW)	SDIs, e.g., Data.gov and Global Earth Observation System of Systems (GEOSS).

In terms of data portals and web services, one of the most critical problems is that users may not know every data portal and web service that contain the resources they need. For example, if a user wants to collect satellite images for land cover change research, to retrieve all the satellite images in the area of interest, the user needs to know every data portal and service that provide satellite images. Similarly, for the catalog approaches, users cannot make sure that they have looked for the data they want from every catalog service as no user knows every catalog service. Another critical problem of the catalog approach is that data providers usually have no incentive to spend time and effort to register their data or services to one or more catalog services. As a result, there is no single catalog service that can provide complete GeoWeb index.

Besides the problems mentioned above, for the data portals/services and catalogs that do not follow open standards, there are also some heterogeneity problems. For example, the user interfaces and procedures between each portal/service or catalog are largely different, as shown in Figure 2. Users need to learn the procedures to access data from different portals. Heterogeneities also exist in data models and communication protocols. Data provided by non-open-standard portals/services and catalogs use proprietary data models and protocols. Users need to spend more time and effort to customize connectors to retrieve and understand the data.

In general, these problems of existing GeoWeb resource discovery approaches make geospatial data discovery inefficient and limit the growth of the GeoWeb. To address this issue, this study learns from the WWW resource discovery solutions as WWW also has wildly-distributed resources. To be specific, we aim at building a GeoWeb search engine to provide users a one-stop service to discover geospatial data while avoiding the requirement of asking data providers to register resources. However, to achieve this GeoWeb search engine vision, one of the first steps is to collect a complete index of GeoWeb. As general web search engines apply web crawlers to collect indexes of web pages, a GeoWeb search engine also requires a GeoWeb resource crawler. Hence, we review the existing GeoWeb resource crawler solutions in the next sub-section.

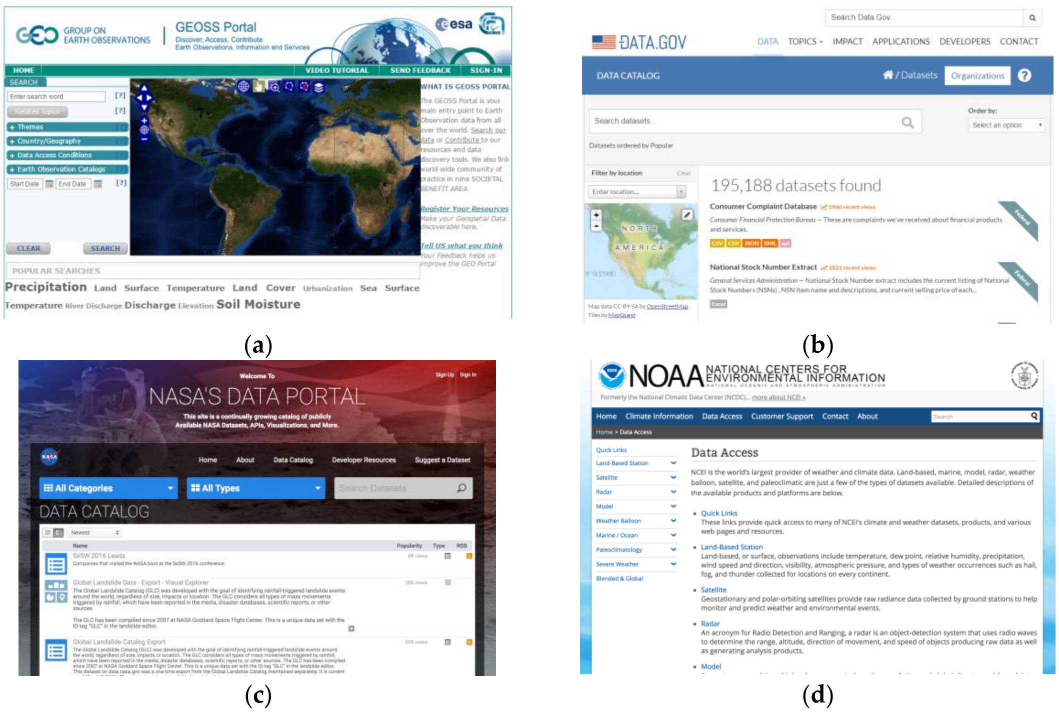


Figure 2. Examples of data portals/SDIs: (**a**) Global Earth Observation System of Systems (GEOSS); (**b**) Data.gov; (**c**) National Aeronautics and Space Administration (NASA)'s data portal; and (**d**) National Climatic Data Center.

2.2. Existing GeoWeb Resource Crawler Solutions

Early web crawlers can be traced back to 1993, including the World Wide Web Wanderer, Jump Station, World Wide Web Worm, and Repository-Based Software Engineering (RBSE) spider that were proposed to collect information and statistics about the Web [18]. Nowadays, one of the main applications of web crawlers is exploring web pages and indexing those pages automatically for web search engines, such as Google and Yahoo! A web crawler uses a list of URLs as seeds to initiate the crawling process. The crawler connects to every seed (i.e., web page), identifies all the hyperlinks in these seeds, and then links to other web pages via those hyperlinks. This process iterates until the crawler crawls all linked web pages on the WWW. With this web crawling approach, web search engines can provide a complete index of web pages to users.

Our research is based on the same web crawling concept, and aims at developing a GeoWeb resource crawler to discover geospatial resources on the Web, such as the OGC web service (OWSs), Keyhole Markup Language (KML) files, and Environmental Systems Research Institute, Inc (ESRI) Shapefiles. According to [19], the first proposed OWS crawler was the Spatial Information Search Engine (SISE) developed by [20]. The NSF-funded PolarHub project (http://polar.geodacenter.org/polarhub2/) also developed an OWS crawler and focused on semantic analysis [21]. Bone et al. [22] built the Geospatial Search Engine (GSE) (http://www.wwetac.us/GSE/GSE.aspx) that provides the OGC Web Map Service (WMS), ArcIMS, ArcGIS Server, and Shapefile. Bone et al. also discussed the functionalities of the search engine such as searching and ranking. Other works include [23] that created a WMS crawler to enhance the US Navy's geospatial information database, [24] that designed a module to finding OGC Web Feature Service (WFS), Web Coverage Service (WCS) and CSW, [25] that focused on efficient crawling for WMS, [26] that developed an OGC-focused crawler and emphasized the statistics of OWS, [27] that crawled for WFS and focused on retrieving semantic annotations of data source by ontology, and [28] that built a specialized search engine to discover ESRI Shapefiles.

These existing works developed GeoWeb crawlers for geospatial resources discovery and had different focuses, such as the crawling efficiency, developing GeoWeb search engine, semantic analysis, etc. While each of these existing GeoWeb resource crawling approaches mainly focused on a subset of GeoWeb resources, there is still a large portion of GeoWeb resource has not been crawled, such as KML or non-open-standard resources. Therefore, in this study, we try to develop an extensible and scalable crawling framework to find various types of GeoWeb resources. To further explain and define the scope of this study, the next section introduces the GeoWeb resource distribution on the WWW.

3. GeoWeb Resource Distribution and Research Scope

To have a more clear definition of our search targets, we classify the WWW into four parts: (1) geospatial web service on the Surface Web; (2) geospatial data on the Surface Web; (3) geospatial web service on the Deep Web; and (4) geospatial data on the Deep Web. The concept is shown in Figure 3. Li [21] defines the "Surface Web" and the "Deep Web" on the WWW. The Surface Web means the Web that users can access through hyperlinks and is visible to general web crawlers. However, some resources are hidden on the WWW, whose addresses cannot be accessed through the Surface Web via hyperlinks. These resources are called in the Deep Web, such as datasets in web services or data portals/SDIs. In order to find resources in the Deep Web, we need to traverse through the services on the Surface Web by following their specific communication protocols.

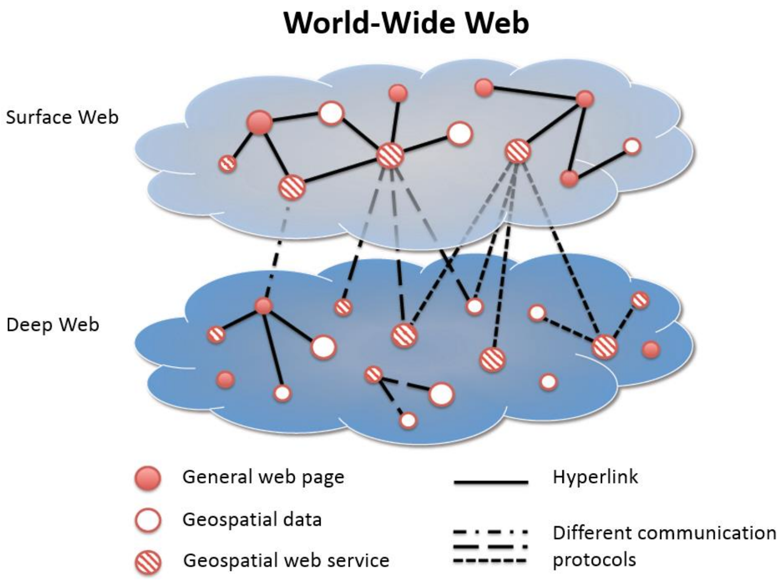


Figure 3. The concept of the Surface Web and the Deep Web.

As aforementioned, to identify and access geospatial resources on the WWW, open standards are necessary since they regulate the communications between clients and servers. If providers share resources without following any open standard, there will be an interoperability problem that users cannot identify or access these resources in a general way. Therefore, with open standards, we can design different identification mechanisms to find resources following these standards, and further parse and index the resources.

While there are some open geospatial web service standards such as the OSGeo TMS, OGC WMS, WFS, and WCS, there are standards for data formats such as the ESRI Shapefile, TIFF/GeoTIFF, GeoJSON, OGC Geography Markup Language (GML) and KML. On the other hand, as aforementioned, geospatial resources may be indexed in catalog services, such as OGC CSW that follow the open standard or SDIs that use proprietary communication protocols.

As web crawlers could discover geospatial resources with identification mechanisms designed accordingly to standards, for the resources that do not follow standards, they could also be discovered but with extra cost to develop customized connectors. While there various types of geospatial resources, a GeoWeb crawling framework should be extensible to identify different resource types. To prove the concept, the searching targets in this study are shown in Table 2. For the data portals and services following open standards, we focused on the OGC WMS, WFS, WCS, Web Map Tile Service (WMTS), Web Processing Service (WPS), and Sensor Observation Service (SOS). These OWS provide maps (e.g., satellite imagery), vector data (e.g., country boundaries), raster data (e.g., digital elevation model), map tiles (e.g., image tiles of an aerial photo), geospatial processing (e.g., buffering process), and sensor observations (e.g., time series of temperature data), respectively. Users need to use specific operations defined in each OWS to retrieve geospatial data from services. While in terms of non-open-standard resources, we select the National Space Organization (NSPO) Archive Image Query System as the target, which provides search functionalities for FORMOSAT-2 satellite images. For catalogs, the OGC CSW and the US data.gov are our open-standard and non-open-standard targets, respectively. In addition, this research also targets at geospatial data formats, including KML files and ESRI Shapefiles. KML is an OGC standard as well, which is a file format for storing geographic features such as points, polylines, and polygons and their presentation styles. KMZ is the compressed KML file that is also one of our search targets. ESRI Shapefile is a popular geospatial vector data format. Although this paper did not target every possible GeoWeb resource types, the proposed solution still provides a larger variety of resources than what any single existing solution provides.

Table 2. GeoWeb resource searching targets.

	Open-standard		Non-open-standard
Data portal/service	1. OGC WMS 2. OGC WFS 3. OGC WCS	4. OGC SOS 5. OGC WMTS 6. OGC WPS	NSPO Archive Image Query System
Catalog	OGC CSW		Data.gov
Files	OGC KML		ESRI Shapefile

4. Methodology

In this section, we present the proposed web crawling framework, GeoWeb Crawler, for discovering geospatial resources on the Web. In general, we considered four challenges in the GeoWeb Crawler. Firstly, as geospatial resources may not be able to directly access with hyperlinks, resources are sometimes presented in the plain-text format on web pages. To discover these plain-text URLs, we use regular expression string matching to find potential URLs instead of only following the HTML links. Secondly, as geospatial resources are in various types, to identify targeted geospatial resources, specific identification mechanisms for each resource type should be designed. Thirdly, crawling resources on the Web is a time-consuming process. In order to improve the efficiency of web crawling, the proposed framework is designed to be scalable with the distributed computing concept. Finally, while the web crawling process traverses through every URL on web pages, identical URLs could be found on different web pages. In this research, we apply the Bloom filter [29] to filter out the redundant URLs. In the following sub-sections, we first introduce the overall workflow of the GeoWeb Crawler, and then present the solutions for the four challenges.

4.1. Workflow

As mentioned earlier, to initiate a web crawler, crawling seeds are required. However, instead of selecting seeds manually and subjectively, we utilize the power of general web search engines. In this research, we use the Google search engine. As Google has already indexed the web pages in the Surface Web, we can directly retrieve relevant web pages by sending designed keywords to the web search engines. In addition, as the Google search engine ranks search results based on the relevance to the request keyword [30], Google could provide us a list of candidate URLs that have a high possibility of containing the target geospatial resources. Therefore, by utilizing the indexing and ranking mechanisms of Google search engine, we could find geospatial resources with fewer crawling requests.

As our goal is to discover geospatial resources that follow OGC service standards or specific data formats, we can design different search keywords, such as "OGC SOS", "web map service", "WFS", and "KML download", to enrich our crawling seeds. In addition, by utilizing the power of existing web search engine, the GeoWeb Crawler can easily extend its search resource targets by simply defining the identification mechanisms and search keywords. In this study, for each keyword search, we used the URLs in the top 10 Google search pages as our crawling seeds.

Figure 4 is the high-level architecture of the GeoWeb Crawler. After retrieving the crawling seeds, the GeoWeb Crawler first sends HTTP GET requests to download HTML documents from the seed web pages. While many hyperlinks are presented in the HTML anchor elements <a>, the crawler extracts the URLs from the attribute "href" in the anchor elements. Then the crawler filters the URLs through some simple constraints, for instance, remove the URLs that end with ".doc", ".pdf", ".ppt", etc. However, URLs may be presented as plain texts on web pages. In order to discover these URLs, we use regular expression string matching (discussed in Section 4.2). Finally, all the valid URLs will be identified if they belong to targeted catalog services, which is the OGC CSW in this research. If true, the crawler will link to the catalog services to find geospatial resources and save the resources into the database. Otherwise, the crawler will examine if URLs belong to targeted geospatial resources according to the identification mechanisms (discussed in Section 4.3). For those URLs that do not belong to any geospatial resource, we will link to the URLs and download their HTML documents. The process will be iterated until all the web pages within a crawling level are crawled. In this study, we set the crawling level to 5, which means that the GeoWeb Crawler can find all the targeted resource within five link-hops from the seed pages. By "link-hop" we mean linking from one web page to another web page. This crawling level is set according to our preferred testing crawling time with our available machines. However, the crawler could find more resources with more link-hops, and the proposed solution certainly can be scaled out to crawl a larger coverage with more machines and Internet bandwidth. Since we only crawled five levels from the seeds, when the crawler reaches the fifth level, the crawler will stop the crawling process. While the five link-hops was set only because of the hardware limitation, the flowchart was to delineate the complete crawling process, which should contain a loop to continuously crawl new web pages.

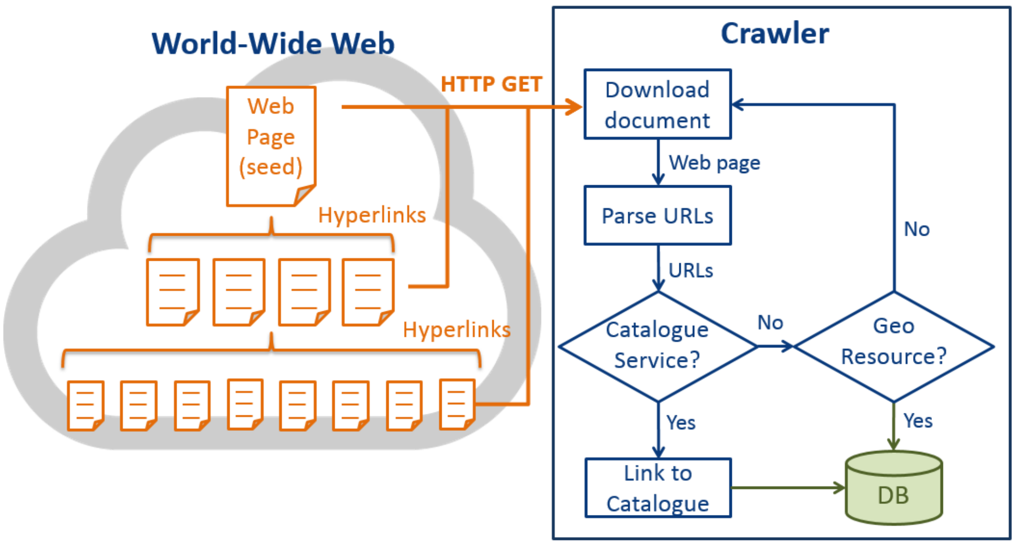


Figure 4. The architecture of the GeoWeb crawler.

The above workflow is for crawling the resources following open-standards. In terms of the non-open-standard resources, the GeoWeb Crawler needs to customize crawlers to a certain degree, which largely depends on each resource. For example, to crawl the resources from the data.gov, we directly set the crawling seeds by keyword-searching in the data.gov. While for the NSPO Archive

Image Query System (http://fsimage.narlabs.org.tw/AIQS/imageSearchE), since the system was designed to be used by humans, which is not crawler-friendly, we used the Application Programming Interface (API) of the data service behind the system to create a customized connector and retrieve FORMOSAT-2 satellite image metadata periodically.

4.2. Discovering Plain-Text URLs with Regular Expressions

As we mentioned earlier, some URLs are not shown in the HTML hyperlink form, but are shown as plain texts, especially for the geospatial resources that can only be accessed via specific communication protocols, such as the OWS. Therefore, we need to parse the plain texts in web pages to retrieve potential URLs.

In order to discover any potential URLs, we apply the regular expression [31]. Since we want to find plain-text URLs on web pages, we design the following regular expression pattern “\ b(https?)://[-a-zA-Z0-9+&@#/%?=~_|!:,.;]*[-a-zA-Z0-9+&@#/%=~_|]”, which is able to discover URLs starts with “http” or “https” in an HTML document.

4.3. Identification Mechanisms for Different Geospatial Resources

With standards, data or services can be interoperable with anyone following the same standard. While geospatial resources have a large variety in terms of data models, communication protocols, and file formats, we need to design different identification mechanisms according to different standards. According to the search targets specified in Table 2, here we discuss the proposed identification mechanisms as follows.

While OGC has defined many open standards for geospatial resources, we choose some of them as the search targets. We first discuss the OGC web services. Most of the OWSs support a mandatory operation called “GetCapabilities” for advertising the metadata and data content of these services. This design provides us an easy way to identify OWSs. By sending GetCapabilities request to every candidate URL, if the URL returns a valid response, the URL belongs to an OWS.

To send a GetCapabilities request, the HTTP GET with the KVP encoding is used during the identification. There are some common parameters in the GetCapabilities operation that could be useful for identifying resources, including “service” that can be used to identify the type of OWS, “request” to specify the GetCapabilities operation. An example of a GetCapabilities request for a WMS is: http://hostname:port/path?service=WMS&request=GetCapabilities.

To identify other types of OWS, we can simply replace the value of the “service” parameter. In general, for every candidate URL, GeoWeb Crawler appends the necessary query options to send the GetCapabilities request. If the URL is an OWS, the crawler will receive a valid Capabilities document from the service. Therefore, we need to design rules to further identify if a response is a valid Capabilities document for the targeted resource.

While Capabilities documents are in the XML format, the document shall contain a specific element name (e.g., “Capabilities”) if the URL is an OWS. However, different versions of different types of OWSs may have different element names. In this case, we need to define rules to verify the Capabilities documents for different OWS and different versions:

- OGC WMS: According to the WMS specification [32,33], the Capabilities document shall contain a root element named “WMT_MS_Capabilities” or “WMS_Capabilities” for the version 1.0.0 and 1.3.0 of WMS, respectively.
- OGC SOS: According to the SOS specification [34,35], the Capabilities document shall contain a root element named “Capabilities” for the version 1.0.0 and 2.0 of SOS.
- OGC WFS: According to the WFS specification [36,37], the Capabilities document shall contain a root element named “WFS_Capabilities” for the version 1.0.0 and 2.0 of WFS.
- OGC WCS: According to the WCS specification [38–40], the Capabilities document shall contain a root element named “WCS_Capabilities” for the version 1.0.0 and “Capabilities” for the version 1.1 and 2.0 of WCS.

- OGC WMTS: According to the WMTS specification [41], the Capabilities document shall contain a root element named "Capabilities" for the version 1.0.0 of WMTS.
- OGC WPS: According to the WPS specification [42], the Capabilities document shall contain a root element named "Capabilities" for the version 1.0.0 and 2.0 of WPS.

In addition, regarding geospatial data files such as the KML and ESRI Shapefile, since these files usually exist as download links on web pages, the identification rules are different from the rules for web services:

- OGC KML: Since KML is a data format, the file extensions would be ".kml" or ".kmz" (i.e., zipped .kml file). So if a URL string ends with ".kml" or ".kmz", we regard it as a KML file.
- ESRI Shapefile: According to the ESRI Shapefile standard [43], a single Shapefile should at least contain a main file (i.e., including ".shp"), an index file (i.e., ".shx") and a database file (i.e., ".dbf"). Therefore, Shapefiles are often compressed as ZIP files when sharing on web pages. The only way to identify if a ZIP file contains a Shapefile is by downloading and extracting the compressed file. Then if the extracted files/folders contain ".shp" files, the URL is regarded as a Shapefile resource. Due to our hardware limitation, we only download and examine the ZIP files that are smaller than 10 megabytes to prove the concept.

Furthermore, for catalog services, as the OGC CSW is also an OWS, CSW can be identified by sending GetCapabilities request.

- OGC CSW: According to the CSW specification [44], the XML document shall contain a root element named "Capabilities" for the version 1.0.0 and 2.0 of CSW.

As CSW serves as a catalog service hosting the indexes of geospatial resources, we can only retrieve the resources by following the communication protocols of CSW. In this case, the *GeoWeb Crawler* uses the "GetRecords" operation to retrieve the resources hidden in the catalog service. Besides the CSW, the proposed GeoWeb Crawler can also crawl the geospatial resources in non-open-standard catalogs. However, as most of the non-open-standard catalogs are not crawler-friendly, data.gov is the one that is crawlable. We assign keywords of different resources to search in the data.gov. Then we obtained a list of candidate web pages as the crawling seeds to initiate the crawling process.

Finally, the resource crawling of the NSPO Archive Image Query System (http://fsimage.narlabs.org.tw/AIQS/imageSearchE) is different from other sources. As the system is not crawler-friendly, we construct a customized connector according to its backend data service interface to periodically retrieve metadata of new FORMOSAT-2 satellite images. In general, by linking to the Deep Web, we can discover the hidden resources to provide a more comprehensive geospatial resources index.

4.4. Distributed Crawling Process for Scalability

To improve the performance of GeoWeb Crawler, we apply the MapReduce concept [17] to execute the crawling process in parallel. The MapReduce concept was proposed by Google to address scalability issues such as calculating indexes in the Google search engine. In general, MapReduce is a processing framework containing two phases, i.e., the *map* phase and the *reduce* phase. In the *map* phase, input data are split into several independent parts. All the parts can be processed in parallel and produce partial results. In *reduce* phase, all the partial results produced in the *map* phase are merged as the final output. MapReduce processes can be executed consecutively to iterate the process until all the tasks are finished.

In general, we design the GeoWeb Crawler with the MapReduce concept. As shown in Figure 5, one machine is assigned as the *master* and connects to a number of *worker* machines. Each *worker* sends a request to the *master* to retrieve candidate URLs as tasks. *Workers* then identify if those URLs belong to geospatial resources with the rules presented in Section 4.3. If a URL is not a geospatial resource, the *worker* will retrieve and parse the HTML file for new URLs. Identified geospatial resources and new URLs are then returned to the *master*. The *master* will store geospatial resources to a database and assign new tasks (i.e., URLs) to *worker*s. As the crawling processes are independent, each *worker* can perform in parallel. With the designed processing workflow, the GeoWeb Crawler can be scaled horizontally to enhance the performance.

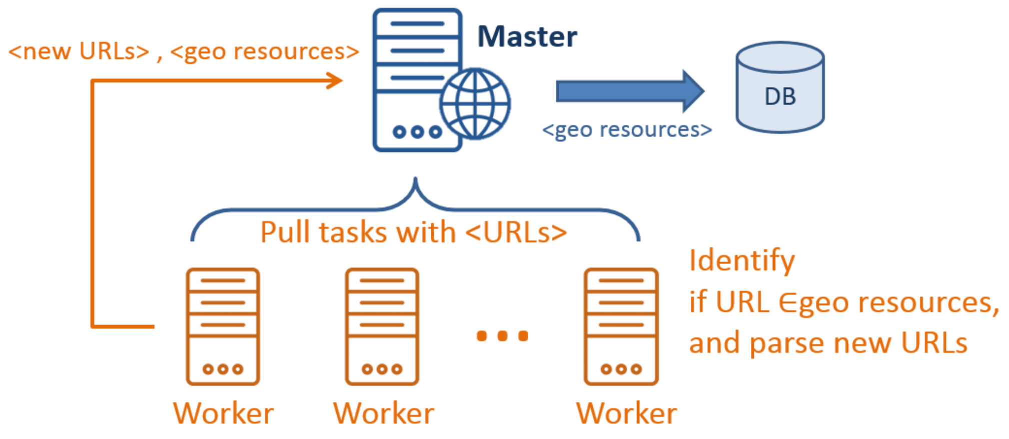


Figure 5. Architecture of distributed computing implementation.

4.5. Filtering Redundant URLs

During the crawling process, a lot of redundant URLs would be found. For example, by using "OGC SOS" as the search keyword, the GeoWeb Crawler totally crawled 9,911,218 URLs within fvie link-hops from the seed pages. However, 7,813,921 of them are redundant URLs. In other words, only 2,097,297 URLs are unique. This huge number of redundancy would seriously affect the crawling performance as the same URLs could be crawled repeatedly. However, recording all the URLs to determine whether they had been crawled or not would be inefficient.

Therefore, in order to deal with this problem, we apply the Bloom filter [29] to identify redundant URLs. The concept of Bloom filter is to transfer string into a binary array, which can keep storage small. As shown in Figure 6, first create an m (size of BitSet) BitSet, and set all bits to 0. Then choose k different hash functions for managing n (number of URL strings) strings. A hash function is any function that can be used to map digital data of arbitrary size to digital data of fixed size, which is called hash values. In this case, we would derive k integers from k hash functions ranging from 0 to $m-1$.

In this study, we transform all characters in a URL string into ASCII codes (integer), then our hash function calculates hash values based on these ASCII codes. Hence, each URL string corresponds to a set of bits determined by hash values. Then set these bits from 0 to 1, as shown in Figure 6. According to the BitSet, we can determine whether two URLs are the same or not. If there is any bit corresponding to a URL is not set to 1, we can conclude that the URL has not been crawled. Otherwise, the URL would be regarded as redundancy.

However, false positive matches are possible, but false negatives are not. Since the size of BitSet is limited, the more elements that are added to the set, the larger the probability of false positives. There is a tradeoff between storage and matching accuracy. Thus, we need to set an appropriate value of m and k according to n. According to [45], assume that $kn < m$, when all the n elements are mapped to m BitSet by k hash functions, the probability of one bit is still 0 (i.e., p′), and the probability of false positive (i.e., f′) is shown in Equations (1) and (2).

$$p' = \left(1 - \frac{1}{m}\right)^{kn} \approx e^{-\frac{kn}{m}} \tag{1}$$

$$f' = \left(1 - \left(1 - \frac{1}{m}\right)^{kn}\right)^{k} \approx \left(1 - \mathrm{p}'\right)^{k} \tag{2}$$

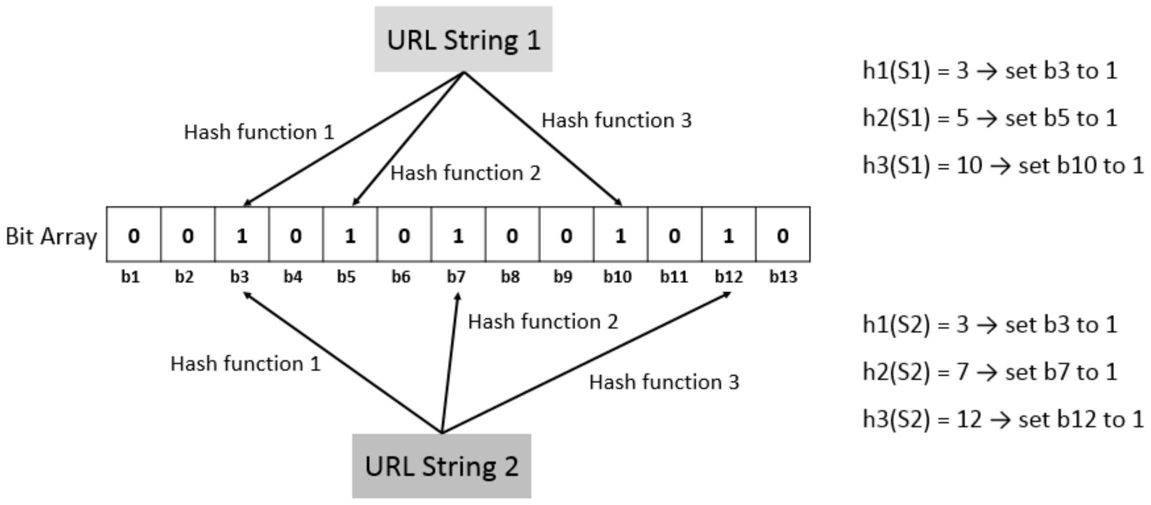


Figure 6. The concept of Bloom filter.

For example, if we set 10 hash functions, and the BitSet size is 20 times larger than the number of URLs, the probability of false positives would be 0.00889%. In other words, about nine errors would happen when handling 100,000 URLs. Even so, such matching quality is still acceptable in this study. In general, with the Bloom filter, we can remove redundant URLs with smaller storage footprint.

5. Results and Discussion

In this section, we present the crawling result and evaluations in different perspectives. Firstly, Section 5.1 shows the latest number of resources discovered by GeoWeb Crawler. Secondly, we compare the numbers of geospatial resources between GeoWeb Crawler and other existing approaches. Thirdly, to better understand the crawling efficiency, we analyze the distribution of discovered resources based on different crawling levels (i.e., link-hops). Finally, since we apply the distributed computing concept to execute the crawling process in parallel, we evaluate the crawling performance by using different numbers of machines.

5.1. Geospatial Resources Crawling Results

In this paper, we design and implement a GeoWeb crawling framework, GeoWeb Crawler, to collect geospatial resources. While the GeoWeb Crawler is designed to be extensible, this paper targeted at various resources, and the crawling result is shown in Table 3. Totally, we discovered 7351 geospatial services and 194,003 datasets within five hyperlink-hops from the seeds. Among these datasets, 1088 service and 77,874 datasets are derived from data.gov and NSPO service, which represent the non-open-standard data services and catalogs in this study.

As web services may have a large amount of geospatial data, we define the idea of the dataset for each service type and show the number of datasets in Table 3. To be specific, for OWSs, we parse the Capabilities document and count the number of datasets.

Basically, we define that a map image is represented as the "Layer" in WMS, a vector dataset is the "FeatureType" in WFS, a raster dataset is the "CoverageSummary" or "CoverageOfferingBrief" in WCS, a map tile image is the "TileMatrix" in WMTS, geospatial processing is the "Process" in WPS, and sensor observation dataset is the "ObservationOffering" in SOS. Although these definitions may not be most appropriate (e.g., "ObservationOffering" as the number of datasets which actually consists of a group of observations), extracting the datasets information from the Capabilities documents prevents us from retrieving actual data from services. Since WMS is a well-known service in OGC, both the number of discovered WMS and the number of datasets are the highest among the OWSs. Regarding the average crawling time, the time mainly depends on the number of crawled URLs and the Internet connection between client and server. As the identifications for files are relatively simple, the crawling time for KML and Shapefile are much lower than that for GeoWeb services.

Table 3. GeoWeb crawling result.

	Number of Services	Number of Datasets	Definition of Dataset	Avg Crawling Time (hr/Keyword)
WMS	4315	45,922	Layer	39.1
SOS	1227	14,875	Observation Offering	32.3
WFS	663	18,930	Feature Type	15.5
WCS	639	12,379	Coverage Summary/Coverage Offering Brief	32.5
WMTS	365	10,060	Tile Matrix	21.4
CSW	117	1121	Record	39.2
WPS	25	2771	Process	14.3
KML	-	4002	kml/kmz file	5.7
Shapefile	-	7878	shp file	2.7
Satellite Image	-	76,065	Image index	-

5.2. Comparison between GeoWeb Crawler and Existing Approaches

There are some existing catalog services that provide geospatial resources. As mentioned in the Introduction, data providers need to register their resources to catalog service to share the resources. However, there is no incentive for providers to register their resources into every catalog. This problem causes that different catalogs provide different resources, and none of them is comprehensive. To address this issue, we use the concept of the web crawler to discover various geospatial resources. Moreover, data providers do not need to register resources as long as they publish resources following standards and share it on the Surface Web.

In general, Figure 7 shows the datasets comparison between GeoWeb Crawler and existing solutions, where the 76,065 satellite images found in the NSPO service is not included in the figure as none of the compared approaches provide this resource. As expected, the number of datasets and the variety of resources that GeoWeb Crawler provides is much higher than that of other approaches. Especially, one thing to note is that the datasets of other compared solutions could be over-estimated. For example, when we search for "WMS" in GEOSS using the keyword "WMS", we directly use the number of total results while not every candidate is WMS, as shown in Figure 8.

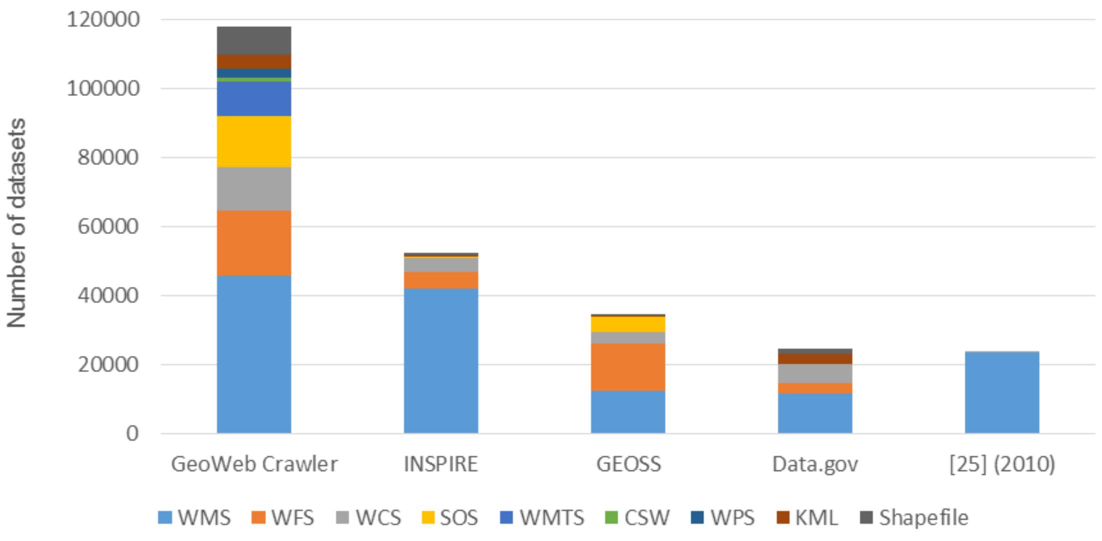
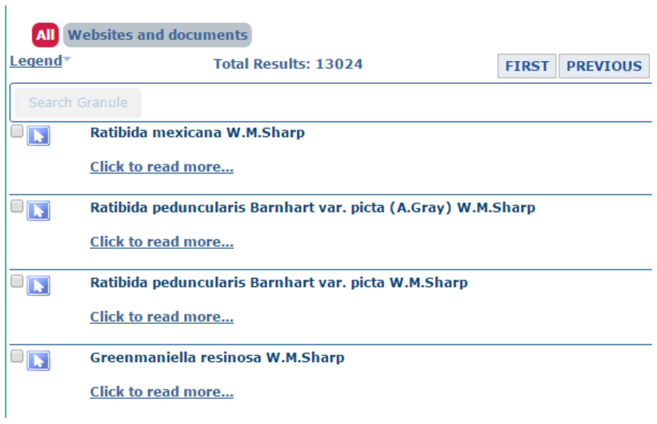


Figure 7. Number of datasets comparison between GeoWeb Crawler and existing approaches.



All Websites and documents

Legend Total Results: 13024 FIRST PREVIOUS

Search Granule

Ratibida mexicana W.M.Sharp

Click to read more...

Ratibida peduncularis Barnhart var. picta (A.Gray) W.M.Sharp

Click to read more...

Ratibida peduncularis Barnhart var. picta W.M.Sharp

Click to read more...

Greenmaniella resinosa W.M.Sharp

Click to read more...

Figure 8. An example of searching WMS in GEOSS.

In addition, there are existing approaches crawling OWS, such as PolarHub [21] in 2014, Sample et al. [23] in 2006, Li and Yang [25] in 2010, and Lopez-Pellicer et al. [26] in 2010. In Figure 9, we compare the numbers of OWS between these works and the GeoWeb Crawler. We can see that the GeoWeb Crawler can collect more types of OWS resources than other work did. With the large-scale crawling framework, GeoWeb Crawler can discover significantly more OWSs than other approaches. Overall, the GeoWeb Crawler collects 3.8 to 47.5 times more resources (including the satellite images) than any existing geospatial resource discovery solution.

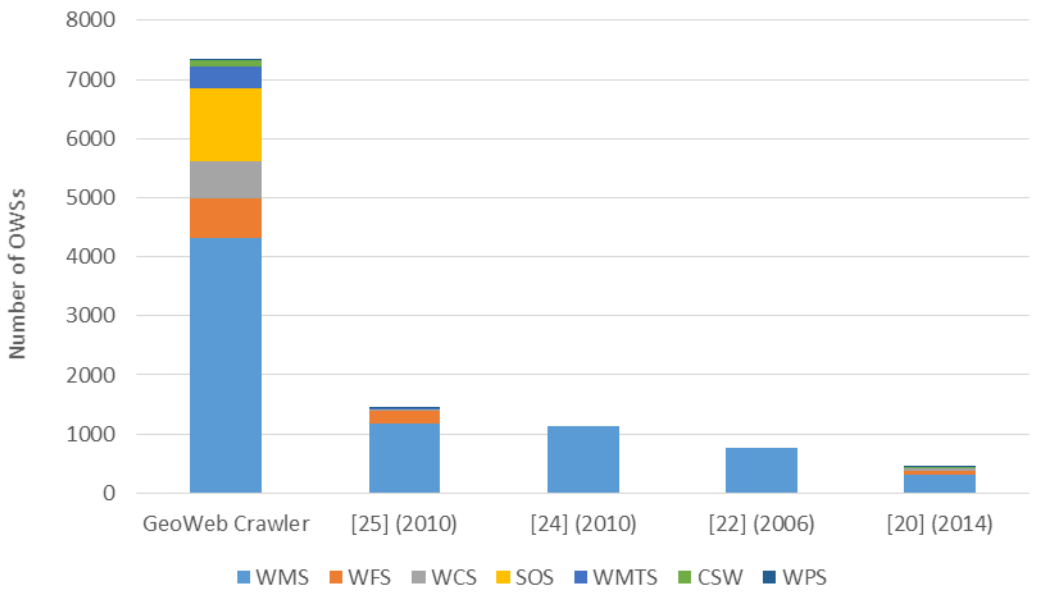


Figure 9. Number of OWSs comparison between GeoWeb Crawler and existing approaches.

5.3. Crawling Level Distribution of Discovered Resources

Since the proposed web crawler cannot crawl the whole WWW because of the limited hardware resource, we design GeoWeb Crawler to be based on the Google search engine. With the indexing and ranking mechanisms of Google, Google search engine can provide us candidates that have a high possibility of containing the geospatial resources with keyword searches. In this case, the *GeoWeb Crawler* could find geospatial resources within fewer link-hops. Currently, we set the crawling level to 5; that is, the crawling scope is all the URLs within fvie link-hops from the Google search result pages. In this study, we adopted breath-first visiting strategy to crawl every complete level one by one.

To understand the distribution of geospatial resources, we display the numbers of discovered resources (services and files) in each crawling level in Figure 10. The figure indicates that more resources can be found with larger crawling level for most of the resources. However, in Figure 11, we show the discovery ratio of each resource in each crawling level, which is the number of discovered resources divided by the total number of URLs in each crawling level. While the discovery ratio represents the necessary effort of crawling, we can see that the trends of discovery ratio are decreasing along the crawling levels because of the rapid growth of URL numbers.

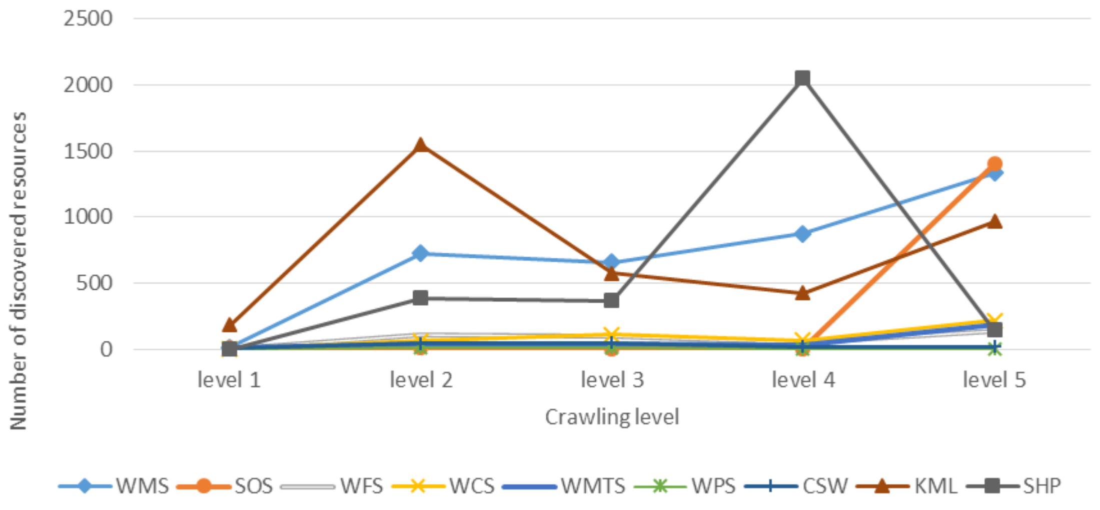


Figure 10. The number of discovered resources in different crawling levels.

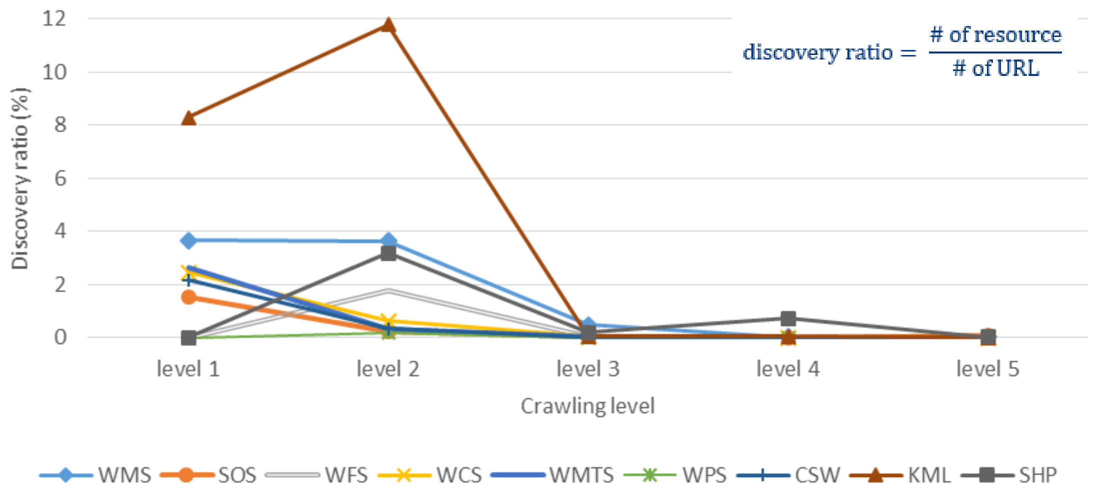


Figure 11. The resources discovery ratio in different crawling levels.

In general, these two figures first indicate that Google search engine could provide fairly good seeds so that we can discover resources with small crawling levels. With the more levels we crawl, the more resources we could discover. However, the discovery ratios also significantly decrease with larger crawling level, which indicates more unnecessary effort spent. Furthermore, in terms of the strange trend behavior of crawling Shapefiles, we believe the reason is that we only examine the ZIP files that are smaller than 10 megabytes to prove the concept, which unavoidably affects the trend behavior. For the peak of KML in level 2, there is a website that provides a tremendous number of KMZ files. Basically, due to the rapid growth of URL numbers along with the crawling level, the number of discovered resources increases correspondingly.

5.4. Crawling Latency Comparison between Standalone and Parallel Processing

In this study, we apply the distributed computing concept to improve the web crawler performance and make the crawling framework scalable. By scalable, we mean that the crawling performance can be improved by simply increasing the number of computers. While the main bottleneck is not on the computation resource but the Internet connection, we argue that the machine specification and machine heterogeneity do not have large influence on the crawling performance. To evaluate the performance of the parallel processing, Figure 12 shows the crawling latencies of using a single machine (i.e., standalone), four machines, and eight machines as *workers* in the proposed crawling framework. We chose to crawl KML files in this evaluation because the identification procedure is the easiest, which can shorten the latency. The crawling of other resources would take much more time (e.g., more than one week) for the standalone setting. As shown in Figure 12, even for crawling KML file, the crawling process still requires about one and half days for the standalone setting. However, with the increasing number of *workers*, the crawling latency can be significantly reduced. This evaluation indicates that the crawling performance is scalable and could be applied to crawl a larger coverage of the Web to discover a comprehensive GeoWeb resource index.

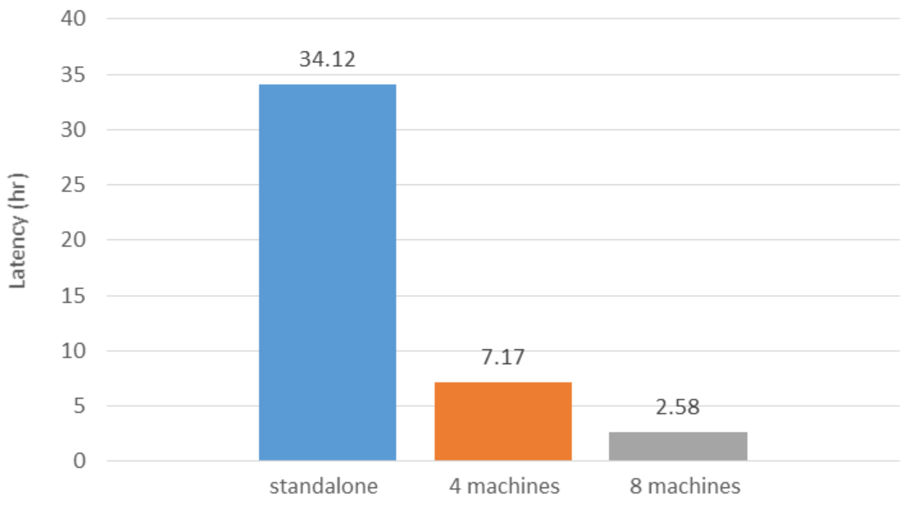


Figure 12. Performance comparison between standalone and parallel processing.

6. GeoHub—A GeoWeb Search Engine Prototype

As the WWW has Web search engines addressing web resource discovery issues, we envision that the GeoWeb also requires GeoWeb search engines for users to efficiently find GeoWeb resources. In this section, we demonstrate the prototype of the proposed GeoWeb search engine, *GeoHub*, which currently allows users to search for OWSs. With the GeoHub user interface, as shown in Figure 13, users can use the keyword search to find the OWSs of their interests. In order to achieve keyword search, the GeoWeb search engine needs a database to manage the metadata that are used as the information for the query operation.

After identifying GeoWeb resources, we store the necessary information into a database. While a search engine could use the various information to rank search results, this prototype simply matches user queries with the following information: the resource URL, resource type, keyword used in Google search engine, content (e.g., Capabilities document), and the source web page. Capabilities documents provide metadata about the OWS. An example of WMS capabilities document is shown in Figure 14. Currently, we also extract specific information from the Capabilities documents, such as service name, title, abstract, keywords, content metadata, etc., and then store them into a database. Hence, whenever users' search keywords exist in the extracted information, the corresponding services will be returned to users (Figure 15).

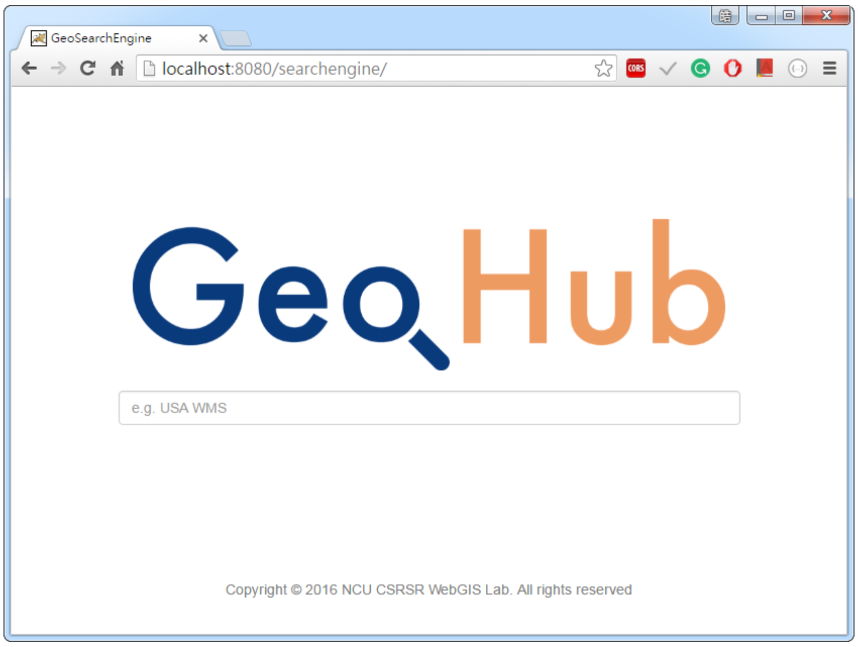
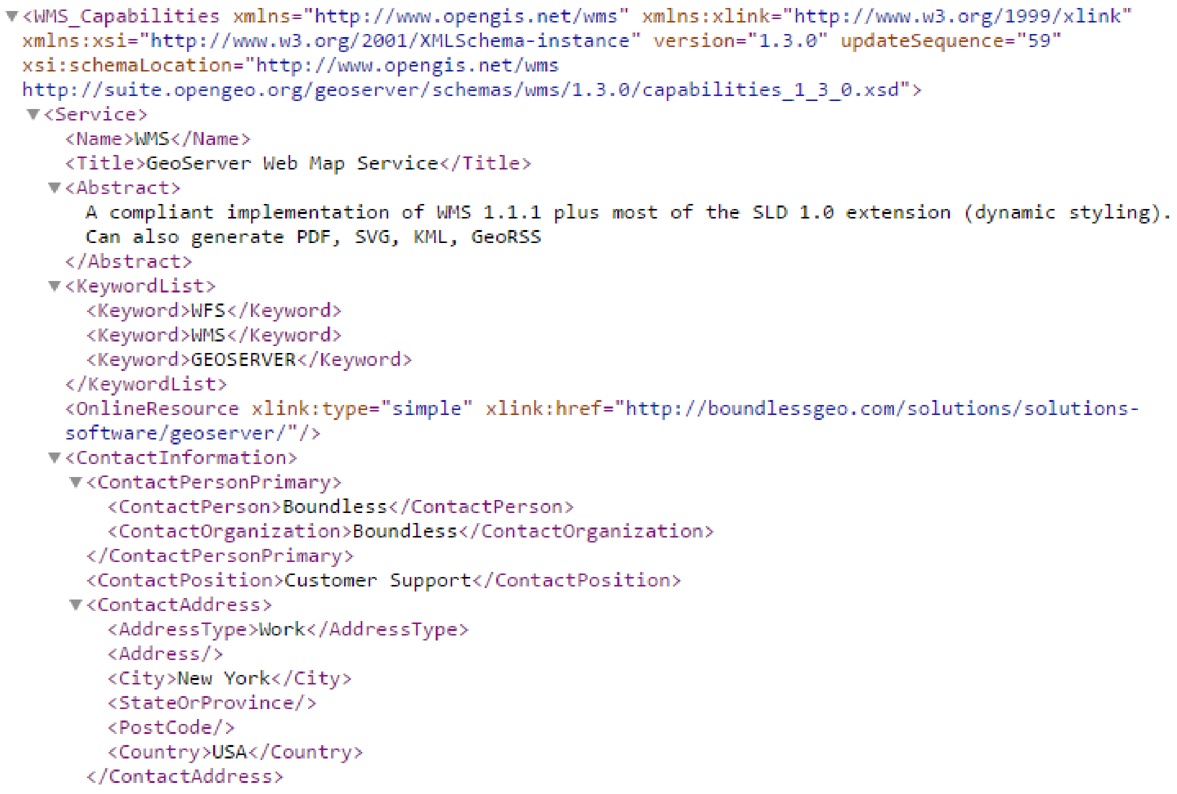
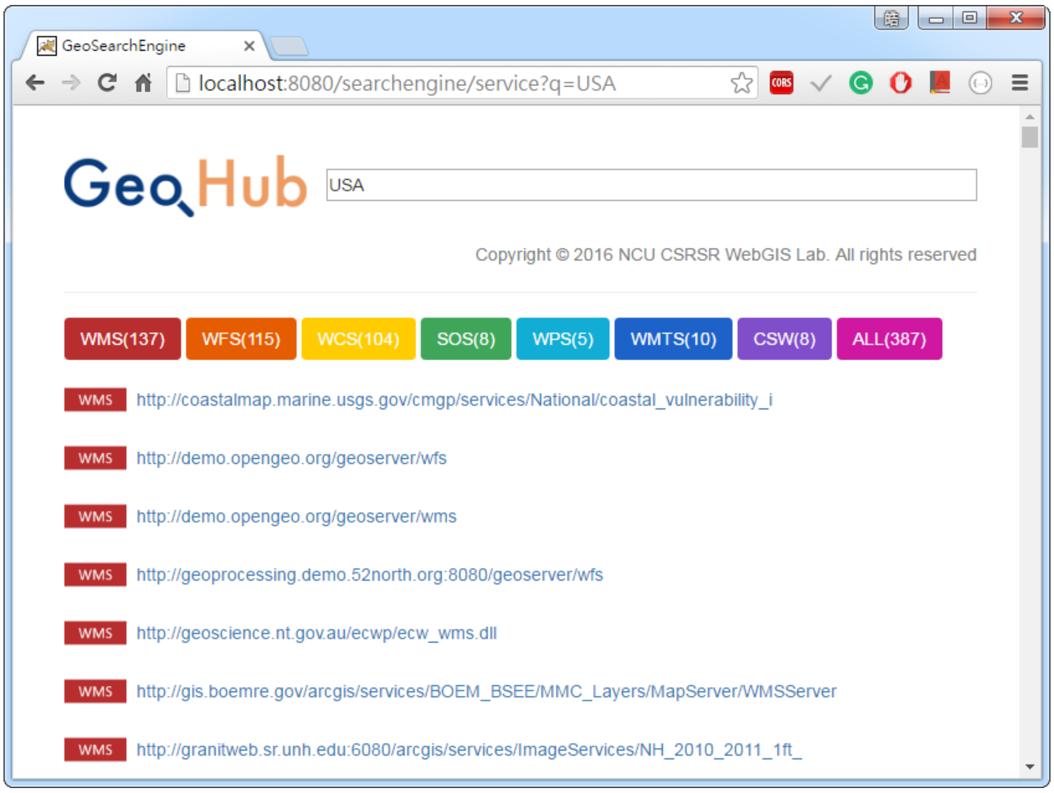

GeoSearchEngine
localhost:8080/searchengine/
GeoHub
e.g. USA WMS
Copyright © 2016 NCU CSRSR WebGIS Lab. All rights reserved

Figure 13. GeoHub client interface.



```
▼<WMS_Capabilities xmlns="http://www.opengis.net/wms" xmlns:xlink="http://www.w3.org/1999/xlink"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="1.3.0" updateSequence="59"
 xsi:schemaLocation="http://www.opengis.net/wms
 http://suite.opengeo.org/geoserver/schemas/wms/1.3.0/capabilities_1_3_0.xsd">
 ▼<Service>
    <Name>WMS</Name>
    <Title>GeoServer Web Map Service</Title>
  ▼<Abstract>
      A compliant implementation of WMS 1.1.1 plus most of the SLD 1.0 extension (dynamic styling).
      Can also generate PDF, SVG, KML, GeoRSS
    </Abstract>
  ▼<KeywordList>
      <Keyword>WFS</Keyword>
      <Keyword>WMS</Keyword>
      <Keyword>GEOSERVER</Keyword>
    </KeywordList>
    <OnlineResource xlink:type="simple" xlink:href="http://boundlessgeo.com/solutions/solutions-
    software/geoserver/"/>
  ▼<ContactInformation>
    ▼<ContactPersonPrimary>
        <ContactPerson>Boundless</ContactPerson>
        <ContactOrganization>Boundless</ContactOrganization>
      </ContactPersonPrimary>
      <ContactPosition>Customer Support</ContactPosition>
    ▼<ContactAddress>
        <AddressType>Work</AddressType>
        <Address/>
        <City>New York</City>
        <StateOrProvince/>
        <PostCode/>
        <Country>USA</Country>
      </ContactAddress>
```

Figure 14. An example of a WMS capabilities document.


GeoSearchEngine
localhost:8080/searchengine/service?q=USA
USA
Copyright © 2016 NCU CSRSR WebGIS Lab. All rights reserved
WMS(137) WFS(115) WCS(104) SOS(8) WPS(5) WMTS(10) CSW(8) ALL(387)

WMS http://coastalmap.marine.usgs.gov/cmgp/services/National/coastal_vulnerability_i
WMS http://demo.opengeo.org/geoserver/wfs
WMS http://demo.opengeo.org/geoserver/wms
WMS http://geoprocessing.demo.52north.org:8080/geoserver/wfs
WMS http://geoscience.nt.gov.au/ecwp/ecw_wms.dll
WMS http://gis.boemre.gov/arcgis/services/BOEM_BSEE/MMC_Layers/MapServer/WMSServer
WMS http://granitweb.sr.unh.edu:6080/arcgis/services/ImageServices/NH_2010_2011_1ft_

Figure 15. An example of a GeoHub search result.

7. Conclusions and Future Work

By using the Web as a platform, the GeoWeb is continuously collecting a tremendous amount of geospatial data. This big geospatial data phenomenon causes difficulties when searching for specific data from such big volume of data with frequent updates and large variety. In order to address this geospatial data discovery issue, our ultimate goal is to build a GeoWeb search engine similar to the existing web search engines. For realizing such GeoWeb search engine, the first step is to discover and index every geospatial data resources on the Web. Thus, in this paper, we design and develop a GeoWeb crawling framework called GeoWeb Crawler. According to different geospatial resource standards, the GeoWeb Crawler can apply different identification mechanisms to discover resources on the Surface Web. For resources on the Deep Web such as OGC CSW and portals/SDIs, the GeoWeb Crawler can support connectors to retrieve the resources. In addition, to improve the performance of web crawlers, we apply the Bloom filter to remove redundant URLs and the distributed computing concept to execute the crawling process in parallel.

According to the experiment result, our proposed approach is able to discover various types of online geospatial resources. Comparing with existing solutions, GeoWeb Crawler can provide comprehensive resources in terms of both variety and amount. Furthermore, instead of requiring data providers to register their data, GeoWeb Crawler can discover resources proactively as long as providers follow standards and share resource links on the Web.

To sum up, as WWW search engines significantly improve the efficiency of web resource discovery, the proposed crawling framework has the potential to become the base of a GeoWeb search engine. GeoWeb search engines could be the next generation of geospatial data discovery framework that can promote the visibility and usage of geospatial resources to various fields and achieve comprehensive data science analysis.

In terms of other future directions, to provide a complete GeoWeb index, we will extend the crawlers to identify other geospatial resource types such as GeoTiff and GeoJSON. We will also seek the opportunity to expand our infrastructure to crawl into the deeper and wider Web.

Acknowledgments: We sincerely thank Ministry of Science and Technology in Taiwan for supporting and funding us to conduct this research.

Author Contributions: Chih-Yuan Huang conceived the research direction; Chih-Yuan Huang and Hao Chang designed the system; Hao Chang develped and evaluated the system; Chih-Yuan Huang and Hao Chang analyzed the result and wrote the paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Densham, P.J. Spatial decision support systems. *Geogr. Inf. Syst. Princ. Appl.* **1991**, *1*, 403–412.
2. Crossland, M.D.; Wynne, B.E.; Perkins, W.C. Spatial decision support systems: An overview of technology and a test of efficacy. *Decis. Support Syst.* **1995**, *14*, 219–235.
3. Burrough, P.A. Principles of geographical information systems for land resources assessment. *Geocar Int.* **1986**, *1*. [CrossRef]
4. Rao, M.; Fan, G.; Thomas, J.; Cherian, G.; Chudiwale, V.; Awawdeh, M. A web-based gis decision support system for managing and planning usda's conservation reserve program (crp). *Environ. Model. Softw.* **2007**, *22*, 1270–1280. [CrossRef]
5. Haklay, M.; Singleton, A.; Parker, C. Web mapping 2.0: The neogeography of the geoweb. *Geogr. Compass* **2008**, *2*, 2011–2039. [CrossRef]
6. Lake, R.; Farley, J. Infrastructure for the geospatial web. In *The Geospatial Web*; Springer: London, UK, 2009; pp. 15–26.
7. Taylor, B. The World is Your Javascript-Enabled Oyster. Official Google Blog. Available online: https://googleblog.blogspot.com/2005/06/world-is-your-javascript-enabled_29.html (accessed on 29 June 2005).
8. Kuhn, W. Introduction to Spatial Data Infrastructures. Presentation held on March 2005. Available online: https://collaboration.worldbank.org/docs/DOC-3031 (accessed on 1 August 2016).
9. Laney, D. *3D Data Management: Controlling Data Volume, Velocity and Variety*; META Group: Terni, Italy, 6 Febuary 2001.
10. Liang, S.H.; Huang, C.-Y. Geocens: A geospatial cyberinfrastructure for the world-wide sensor web. *Sensors* **2013**, *13*, 13402–13424. [CrossRef] [PubMed]
11. Botts, M.; Percivall, G.; Reed, C.; Davidson, J. OGC® sensor web enablement: Overview and high level architecture. In *Geosensor Networks*; Springer: Berlin, Germany, 2006; pp. 175–190.
12. Liang, S.; Chen, S.; Huang, C.; Li, R.; Chang, Y.; Badger, J.; Rezel, R. Capturing the long tail of sensor web. In Proceedings of International Workshop on Role of Volunteered Geographic Information in Advancing Science, Zurich, Switzerland, 14 September 2010.
13. Chris, A. *The Long Tail: Why the Future of Business is Selling Less of More*; Hyperion: New York, NY, USA, 2006.
14. Morais, C.D. Where is the Phrase "80% of Data is Geographic" from? Available online: https://www.gislounge.com/80-percent-data-is-geographic/ (accessed on 28 December 2014).
15. Sullivan, D. Major Search Engines and Directories. Available online: http://www.leepublicschools.net/Technology/Search-Engines_Directories.pdf (accessed on 1 August 2016).
16. Hicks, C.; Scheffer, M.; Ngu, A.H.; Sheng, Q.Z. Discovery and cataloging of deep web sources. In Proceedings of the 2012 IEEE 13th International Conference on Information Reuse and Integration (IRI), Las Vegas, NV, USA, 8–10 Augsut 2012; pp. 224–230.
17. Dean, J.; Ghemawat, S. Mapreduce: Simplified data processing on large clusters. *Commun. ACM* **2008**, *51*, 107–113. [CrossRef]
18. Mirtaheri, S.M.; Dinçktürk, M.E.; Hooshmand, S.; Bochmann, G.V.; Jourdan, G.-V.; Onut, I.V. A brief history of web crawlers. In Proceedings of the 2013 Conference of the Center for Advanced Studies on Collaborative Research, Riverton, NJ, USA, 18 November 2013; pp. 40–54.

19. Lopez-Pellicer, F.J.; Rentería-Agualimpia, W.; Nogueras-Iso, J.; Zarazaga-Soria, F.J.; Muro-Medrano, P.R. Towards an active directory of geospatial web services. In *Bridging the Geographic Information Sciences*; Springer: Berlin, Germany, 2012; pp. 63–79.
20. Bai, Y.; Yang, C.; Guo, L.; Cheng, Q. Opengis wms-based prototype system of spatial information search engine. In Proceedings of the 2003 IEEE International Geoscience and Remote Sensing Symposium, 2003. IGARSS'03, Toulouse, France, 21–25 July 2003; pp. 3558–3560.
21. Li, W. Polarhub: A global hub for polar data discovery. In Proceedings of the AGU Fall Meeting Abstracts, San Francisco, CA, USA, 3–7 Decemebr 2014.
22. Bone, C.; Ager, A.; Bunzel, K.; Tierney, L. A geospatial search engine for discovering multi-format geospatial data across the web. *Int. J. Digital Earth* **2014**, *9*, 1–16. [CrossRef]
23. Sample, J.T.; Ladner, R.; Shulman, L.; Ioup, E.; Petry, F.; Warner, E.; Shaw, K.B.; McCreedy, F.P. Enhancing the us navy's gidb portal with web services. *IEEE Internet Computing* **2006**, *10*, 53–60. [CrossRef]
24. Chen, N.; Liping Di B, G.Y.B.; Chen, Z. Geospatial Sensor Web Data Discovery and Retrieval Service Based on Middleware. 2008. Available online: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.184.4226 (accessed on 1 August 2016).
25. Li, W.; Yang, C.; Yang, C. An active crawler for discovering geospatial web services and their distribution pattern—A case study of OGC web map service. *Int. J. Geogr. Inf. Sci.* **2010**, *24*, 1127–1147. [CrossRef]
26. Lopez-Pellicer, F.J.; Béjar, R.; Florczyk, A.J.; Muro-Medrano, P.R.; Zarazaga-Soria, F.J. State of Play of OGC Web Services Across the Web. In Proceedings of the INSPIRE Conference, Krakow, Poland, 23–25 June 2010.
27. Patil, S.; Bhattacharjee, S.; Ghosh, S.K. A spatial web crawler for discovering geo-servers and semantic referencing with spatial features. In Proceedings of the International Conference on Distributed Computing and Internet Technology, Bhubaneswar, India, 6–9 February 2014; pp. 68–78.
28. Gibotti, F.R.; Câmara, G.; Nogueira, R.A. Geoinfo 2015. In *Geodiscover—A Specialized Search Engine to Discover Geospatial Data in the Web*; GeoInfo: Campos do Jordão, Brasil, 2005; pp. 3–14.
29. Bloom, B.H. Space/time trade-offs in hash coding with allowable errors. *Commun. ACM* **1970**, *13*, 422–426. [CrossRef]
30. Page, L.; Brin, S.; Motwani, R.; Winograd, T. The Pagerank Citation Ranking: Bringing Order to the Web. Available online: http://ilpubs.stanford.edu:8090/422/ (accessed on 1 August 2016).
31. Kleene, S.C. Representation of Events in Nerve Nets and Finite Automata. Available online: www.dtic.mil/cgi-bin/GetTRDoc?Location=U2&doc=GetTRDoc.pdf&AD=ADA596138 (accessed on 1 August 2016).
32. De La Beaujardiere, J. Open GIS® Web Map Server Interface Implementation Specification. Revision 1.0.0. Available online: http://portal.opengeospatial.org/files/?artifact_id=7196 (accessed on 4 August 2016).
33. De La Beaujardiere, J. Open GIS® Web Map Server Implementation Specification. Version: 1.3.0. Available online: http://portal.opengeospatial.org/files/?artifact_id=14416 (accessed on 4 August 2016).
34. Na, A.; Priest, M. Sensor Observation Service. Available online: http://portal.opengeospatial.org/files/?artifact_id=26667 (accessed on 4 August 2016).
35. Bröring, A.; Stasch, C.; Echterhoff, J. OGC Sensor Observation Service Interface Standard. Available online: https://portal.opengeospatial.org/files/?artifact_id=47599 (accessed on 4 August 2016).
36. Vretanos, P.A. Web Feature Service Implementation Specification. Available online: http://portal.opengeospatial.org/files/?artifact_id=7176 (accessed on 4 August 2016).
37. Vretanos, P.A. Opengis Web Feature Service 2.0 Interface Standard. Available online: http://portal.opengeospatial.org/files/?artifact_id=39967 (accessed on 4 August 2016).
38. Evans, J.D. Web Coverage Service (WCS), Version 1.0. 0 (Corrigendum). Available online: https://portal.opengeospatial.org/files/05-076 (accessed on 4 August 2016).
39. Whiteside, A.; Evans, J.D. Web Coverage Service (WCS) Implementation Standard. Available online: https://portal.opengeospatial.org/files/07-067r5 (accessed on 4 August 2016).
40. Baumann, P. OGC WCS 2.0 Interface Standard—Core. Open Geospatial Consortium. Available online: https://portal.opengeospatial.org/files/09-110r4 (accessed on 4 August 2016).
41. Maso, J.; Pomakis, K.; Julia, N. Open GIS Web Map Tile Service Implementation Standard. Available online: http://portal.opengeospatial.org/files/?artifact_id=35326 (accessed on 4 August 2016).
42. Schut, P.; Whiteside, A. Open GIS Web Processing Service, OGC Project Document. Available online: http://portal.opengeospatial.org/files/?artifact_id=24151 (accessed on 4 August 2016).
43. ESRI, U.; PaperdJuly, W. ESRI shapefile technical description. *Comput. Stat* **1998**, *16*, 370–371.

44. Nebert, D.; Whiteside, A.; Vretanos, P. Open GIS Catalogue Services Specification. Available online: http://portal.opengeospatial.org/files/?artifact_id=20555 (accessed on 4 August 2016).
45. Broder, A.; Mitzenmacher, M. Network applications of bloom filters: A survey. *Internet Math.* **2004**, *1*, 485–509. [CrossRef]