International Journal of Geo-Information

MDPI

Article

Reconstructing Sessions from Data Discovery and Access Logs to Build a Semantic Knowledge Base for Improving Data Discovery

Yongyao Jiang [1], Yun Li [1], Chaowei Yang [1,*], Edward M. Armstrong [2], Thomas Huang [2] and David Moroni [2]

[1] NSF Spatiotemporal Innovation Center, George Mason University, Fairfax, VA 22030, USA; yjiang8@gmu.edu (Y.J.); yli38@gmu.edu (Y.L.); cyang3@gmu.edu (C.Y.)

[2] NASA Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA 91109, USA; Edward.M.Armstrong@jpl.nasa.gov (E.M.A.); Thomas.Huang@jpl.nasa.gov (T.H.); David.F.Moroni@jpl.nasa.gov (D.F.M.)

* Correspondence: cyang3@gmu.edu; Tel.: +1-703-993-4742

Academic Editor: Wolfgang Kainz

Received: 1 February 2016; Accepted: 18 April 2016; Published: 25 April 2016

Abstract: Big geospatial data are archived and made available through online web discovery and access. However, finding the right data for scientific research and application development is still a challenge. This paper aims to improve the data discovery by mining the user knowledge from log files. Specifically, user web session reconstruction is focused upon in this paper as a critical step for extracting usage patterns. However, reconstructing user sessions from raw web logs has always been difficult, as a session identifier tends to be missing in most data portals. To address this problem, we propose two session identification methods, including time-clustering-based and time-referrer-based methods. We also present the workflow of session reconstruction and discuss the approach of selecting appropriate thresholds for relevant steps in the workflow. The proposed session identification methods and workflow are proven to be able to extract data access patterns for further pattern analyses of user behavior and improvement of data discovery for more relevancy data ranking, suggestion, and navigation.

Keywords: web usage mining; session identification and reconstruction; crawler detection; semantic search; data discovery

1. Introduction and Literature Review

We are in the era of "Big Data". Spatiotemporal data, which contains both spatial and temporal information, such as satellite imagery captured through remote sensors, climate projections generated from large scale simulations and geospatially tagged social media data, are ever increasing [1]. However, recent advances in remote sensing satellites and other sensors have made spatiotemporal data growing even faster. All these developments are leading to an increase in volume, velocity, and variety of spatiotemporal data and pose a grand challenge to researchers for both discovering and accessing such data for research and decision support applications. In response, a number of Spatial Data Infrastructure (SDI) components (e.g., catalogues and portals) have been developed [2,3]. For example, Common Metadata Repository (CMR) is developed by NASA to enable the science community to more easily use and exchange NASA's data and services. In reality most SDIs are still using a keyword-based search, one of the most rudimentary methods of information retrieval, using the notion of exact matching to match documents to a user query. The keyword-based search inherits the most common information retrieval problems: synonymy and polysemy [4].

Semantic search offers a solution to overcome the problems by understanding users' intent and the contextual meaning of terms using the knowledge base, also called ontology [5]. An effective and robust semantic knowledge base is difficult to build as it is very cost intensive to keep up-to-date as domains change. Meanwhile, the remarkable success of web usage mining in website personalization, system improvement, and business intelligence [6,7], provides the valuable patterns (e.g., keywords and datasets relationship) underneath users' browsing behavior on data portals. In this regard, we propose an approach to mine these patterns to construct a semantic base in a more effective and efficient manner.

Our goal is to mine users' search and usage patterns to improve the data discovery process, which includes mining the knowledge and applying this user knowledge in data discovery. This paper focuses on the prior step of a novel technique for revealing the intent of user searches. Prior to extracting user knowledge (e.g., latent semantic relationship between search terms) from a massive amount of data access logs, a necessary and critical step would be reconstructing users' searching and browsing behaviors by eliminating noise, identifying human users as opposed to machine users or crawlers, as well as connections between each individual request. As an important step, web session reconstruction is the main focus of this paper to lay the groundwork for further analysis. Research on the later steps of what could be done based on the session reconstruction results are also discussed in Section 6.

First, all activities performed by the same person should be grouped together. Second, all activities belonging to the same visit should be placed into the same group [8]. According to W3C (W3C Web Usage Characterization Activity 1999), user session refers to the group of activities performed by a user from the moment he or she enters the site to the moment he or she leaves the site. To keep track of user activities, many websites adopt session identifier in their network communication to ensure the current state of user is not lost when using a stateless protocol such as HTTP. However, in most cases, this piece of information is not available. Several user identification approaches are devised to address this problem, and they generally fall into two categories of time-based heuristics and referrer-based heuristic. Two variations of time-based heuristics and a basic referrer-based heuristic are given below:

- h1: Total session duration may not exceed a threshold α. Given t0, the timestamp for the first request in a constructed session S, the request with timestamp t is assigned to S, if $t - t0 \leqslant \alpha$.
- h2: Total time spent on a page may not exceed a threshold α. Given t1, the timestamp for request assigned to constructed session S, the next request with timestamp t2 is assigned to S, if $t2 - t1 \leqslant \alpha$.
- href: Given two consecutive requests p and q, with p belonging to constructed session S. Then q is assigned to S, if the referrer for q was previously invoked in S.

Traditional time-oriented heuristic based on an empirical and fixed timeout [9] is a significant topic. Commonly used time thresholds for h1 and h2 are 30 and 10 min, respectively [10]. The main argument is that the timeout threshold should be site-specific and depend on website structures and user groups. For instance, Jones and Klinkner found the 30 min threshold performed "no better than random" in the context of identifying search tasks [11]. Some papers mentioned that the threshold could be determined by using website usage statistics [12], but none of them explicitly addressed this in detail. To fill this gap, we present both the theory and practice of selecting an appropriate threshold for time-based heuristics method in this article.

It has been argued that (1) different users' browsing speed varies from one to the other; (2) even for the same user, he or she could spend different times on different pages. Some dynamic heuristic have been introduced to deal with this problem with certain limitations [9]. The idea behind clustering-based heuristic is that session identification could essentially be considered as a clustering problem on one dimension, *i.e.*, time. Driven by this concept, we adopted clustering analysis to automatically detect the session break, rather than using a fixed threshold. However, the number of clusters is usually unknown before the clustering process, which makes it difficult to apply common clustering algorithms such as the k-means clustering method. To avoid assigning a cluster number in the beginning, we developed a

hierarchical clustering method, also known as cluster-based heuristics, to build a hierarchy of clusters on time dimension.

A referrer-based heuristic shows poor performance on sites with framesets due to implicit assumptions about web architecture [8]. The sheer complexity of this strategy and its developmental focus on task over session make it unsuitable as a replacement for time-oriented heuristics in practical web analytics of user sessions [13]. Specifically, the first significant problem is that it only focuses on the task rather than session and tends to generate many short sessions in some cases, where a user performs several tasks in a certain session but starts from the first page for each individual task. On the other hand, there is a chance that several short sessions could be merged into an unreasonably long session when random events such as making phone calls or having lunch breaks happen and the web page is left open during their absence. To address this challenge, we developed a time-referrer-based heuristic by introducing a time threshold in the existing referrer-based heuristic.

Most work in this area presents their session reconstruction result with a simple table [14–16]. This presentation method makes it difficult for users and researchers to analyze the session structure. There is also difficulty in applying and integrating each individual technique because in reality some techniques are performed at different processing stages. For example, crawler detection is partially performed before session identification and the rest is done after session identification. In addition, it is typical that the content served to users comes from multiple servers in a large-scale web system [17]. In order to keep track of users' "inter-site" browsing behavior, it is essential to conduct global synchronization across different servers [18]. In the absence of user and session identifiers, referrer-based heuristic is usually used to build connection between different HTTP logs [19]. However, none of them address the issue of connection between HTTP log and FTP log, which does not contain referrer and user-agent information. Therefore, we describe the workflow of session reconstruction from data import to resulting visualization and also highlight the synchronization of logs from multiple servers.

2. Data Format and Preparation

2.1. Web Log Format

The Common Log Format is the most widely used log format maintained by W3C. It has a number of fields including client IP address, request date/time, page requested, HTTP code, and bytes served (W3C Extended Log File Format, Table 1). Another popular format is the Combined Log Format that is the same as the Common Log Format except with the addition of two more fields: user agent and referrer (Table 1). The user agent is the identifying information that the client browser reports itself. The referrer tells where the request originated [20]. Similarly, properties included in FTP log format are date, transfer-time, remote-host, file-size, filename, transfer-type, special-action-flag, direction, access-mode, username, service-name, authentication-method, authenticated-user-id, and completion-status [21] (Table 2). What makes FTP log different from HTTP log is that FTP log does not contain referrer and user-agent information.

Table 1. Sample HTTP log data in Combined Log Format.

<IP> - - <Date> <Method> <Request> <Protocol> <Code> <Bytes> <Referrer> <User-agent>
68.180.228.99 - - [31/Jan/2015:23:59:13 -0800] "GET /datasetlist/... HTTP/1.1" 200 84779 "-" "Mozilla/5.0 ..."
185.10.104.195 - - [31/Jan/2015:23:59:19 -0800] "GET /datasetlist/... HTTP/1.1" 200 83486 "-" "Mozilla/5.0 ..."
185.10.104.196 - - [31/Jan/2015:23:59:25 -0800] "GET /datasetlist... HTTP/1.1" 200 84357 "-" "Mozilla/5.0 ..."
198.118.243.101 - - [31/Jan/2015:23:59:37 -0800] "GET /dataset/... HTTP/1.0" 200 117223 "-" "gsa-crawler..."

Table 2. Sample FTP log data in FTP Log Format.

<Date > <Transfer-time > <IP > <File-size > <File-name > <Transfer-type >_< Transfer-direction > < Access-mode > < User-name >< Service > < Authentication-method >*< Completion-status >
Mon Feb 16 23:43:29 2015 1 66.249.65.134 698872 /allData/... b _ o a lftp@ ftp 0 * c
Mon Feb 16 23:43:29 2015 1 130.54.59.5 103307 /allData/... b _ o a lftp@ ftp 0 * c
Mon Feb 16 23:43:30 2015 1 130.54.59.5 103455 /allData/... b _ o a lftp@ ftp 0 * c
Mon Feb 16 23:43:30 2015 1 66.249.65.142 168421 /allData/... b _ o a lftp@ ftp 0 * c

2.2. Data Source

To demonstrate the proposed session reconstruction methods, web logs from the Physical Oceanography Distributed Active Archive Center (PO.DAAC; http://podaac.jpl.nasa.gov) website are used. PO.DAAC is a NASA data center, mainly responsible for archiving and distributing oceanographic datasets. Its facility is managed and located at NASA's Jet Propulsion Laboratory (JPL) in Pasadena, California. PO.DAAC has at least two types of servers in their system, with HTTP server providing searching capacity and FTP server supporting downloading request. In this experiment, we use the web logs from February 2015, which includes 4,191,741 PO.DAAC web access logs and 3,174,458 FTP logs.

2.3. User Identification

User identification is the first step in session reconstruction to distinguish among different users for the next step and further data mining. In the absence of user identifier and client-side cookie, the commonly used approach of identifying unique users is through a combination of IP addresses and user agents [22]. IP addresses alone are not sufficient for mapping log entries to the set of unique visitors, mainly because a single proxy server may have several users accessing a website, potentially over the same time period. Given the data center's charter to provide data freely available to the public, data downloads are done anonymously. This means a lack of user agent information in FTP logs. In order to bring FTP site information into our analysis, user agent information is not considered in the user identification.

2.4. Data Cleansing

The goal of data cleansing is to remove redundant logs and data generated by crawlers. In most cases, only the log entry of a HTML file request should be kept for further analysis because a user does not explicitly make all the requests, of which most are automatically downloaded images, JavaScript, and CSS files embedded in a web page. Another common problem is that a typical log file usually contain a significant (sometimes as high as 50%) percentage of references resulting from search engine or other crawlers [17]. In order to distinguish the behavior of web crawlers from actual users, we developed a method to detect crawlers from different aspects [17,23]:

- Well-known search engine crawlers are the easiest to detect, because they usually write their identities in the user-agent field. Therefore, they could be identified and removed by maintaining a list of known crawlers.
- Other "well-behaved" crawlers, which abide by standard robot exclusion protocols, begin their site crawl by first accessing exclusion file "robots.txt" in the server root directory. Such crawlers, can therefore, be identified by checking whether a request to the robots.txt file was made.
- Unfortunately a lot of crawlers neither identify themselves explicitly; nor deliberately masquerade as legitimate users. In this case, we examine other two important features: maximum sustained request rate and the number of request types. The rationale behind this is that there is an upper bound on the maximum number of clicks that a human can make within a specific time frame. Also, after looking into many crawler requests, we found that requests generated by humans are more diverse during their single visit.

3. Methodology for Session Identification

Session identification is the process of segmenting the user activity records of each user into sessions, each record representing a single visit to the site. This section discusses how to select appropriate thresholds for time-based heuristics. We also introduce two new heuristics of clustering-based and time-referrer-based heuristics.

3.1. Threshold Selection in Time-Based Heuristics

We therefore present a method to select appropriate thresholds based on usage statistics. Once user identification and data cleansing is completed, we generate the inter-activity times for each user. Inter-activity time is the time interval between the given request and its last request. According to a recent related research [13], statistically significant patterns exist in inter-activity time if we plot the histogram and component Gaussian mixture model using expectation maximization [24]. As described in [13,24], there are four types of patterns identified by visually inspecting: simple bimodal fits, fits with extended breaks, fits with a high frequency component, and unusual fits. The most common pattern is the bimodal fits: the first curve represents the theoretical within-session cluster with an expected value of several minutes (<10 min), and the second curve refers to the theoretical between-session cluster with an expected value of several days. Inspired by this result, we first segment the inter-activity times into two parts by selecting a relatively large estimated cutoff (e.g., 3 h), and calculate the inter-activity value at the 97.5% confidence level. This value can then be used as α in time-based heuristics.

3.2. Clustering-Based Heuristics

To avoid assigning a cluster number in the beginning, we developed a hierarchical clustering method (i.e., a cluster-based heuristic), to build a hierarchy of clusters on the time dimension. Specifically, this strategy is a divisive or "top down" approach: all observations are seen as one entire cluster, and splits are performed recursively as one goes down the hierarchy. This method consists of three steps (Figure 1):

1. For a given user, his or her visit is thought of as an entire cluster L. If the length of his or her visit is longer than T, all the access logs (sorted by time, including both HTTP and FTP) would be split into two clusters: l_1 and l_2. Threshold T could be considered as the study scale.
2. If the length of l_1 or l_2 is longer than T, they would be split into two clusters again.
3. The split process is performed recursively until all the cluster length is shorter than T.

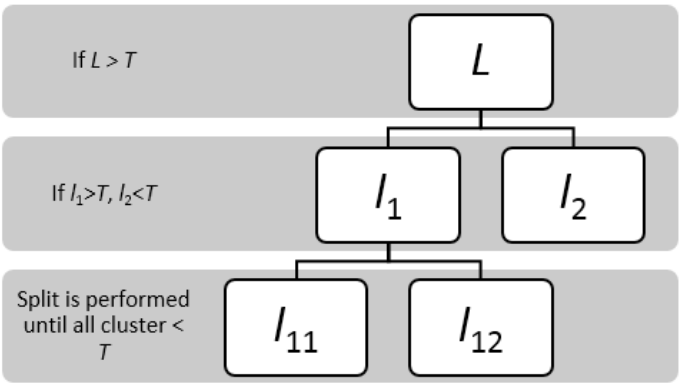


Figure 1. Example of clustering-based heuristics.

In order to decide where a cluster should be split, a method called Jenks natural breaks optimization [25], originally used for statistical mapping, is integrated into our approach. Jenks natural breaks optimization is an iterative process, seeking to find the best break by reducing the variance within classes and maximize the variance between classes. This process is started by dividing the ordered data into groups and allows for initial group divisions to be arbitrary. There are four steps that must be repeated [26]:

1. Calculate the sum of squared deviations between classes (SDBC).
2. Calculate the sum of squared deviations from the array mean (SDAM).
3. Subtract the SDBC from the SDAM (SDAM-SDBC). This equals the sum of the squared deviations from the class means (SDCM).
4. After inspecting each of the SDBC, a decision is made to move one unit from the class with the largest SDBC toward the class with the lowest SDBC.
5. New class deviations are then calculated, and the process is repeated until the sum of the within class deviations reaches a minimal value. Based on Jenks natural breaks optimization, the best break in step two of our clustering-based heuristic could be identified.

3.3. Time-Referrer-Based Heuristics

Considering the referrer (immediately previously accessed) web page in the web log as an important information to keep track of user behavior, we also developed a time-referrer-based heuristic to address the problems in existing referrer-based heuristics. Time-referrer-based heuristics address these two problems by introducing a time threshold, which has already been determined in Section 3.1. It starts by sorting each user's logs by access time, and then goes through the following four steps:

For HTTP logs,

1. If the referrer of log q is "-", URL from other websites (e.g., commercial search engine) or the first page of website, a new session S starts.
2. If the referrer r is none of the three cases in step 1, we would look for the most recent page p whose request is identical to r. Instead of simply assigning log q to session S as they do in traditional referrer-based heuristic, we calculate the time interval T_{pq} between p and q. Then the time interval is compared with $T * N$. Note that N is the number of logs between p and q, and T is the time threshold in the first section.
3. If $T_{pq} < T * N$, log q is assigned to session S. Otherwise, if $T_{pq} > T * N$, or previous page is not found, a new session starts.
4. After all the logs are visited, close sessions are merged together if the time interval between the ending time of one session and the starting time of the other is less than T.

Since FTP logs do not have referrer information:

5. If the time interval from the last log, either HTTP or FTP, is less than T, the FTP log is assigned to the same session and the last log. Otherwise, a new session starts.

Table 3 is an example that demonstrates advantage of the proposed method. The number in the table represents the number of sessions. From the traditional method to the intermediate result of the proposed method, session 1 changes from a long session that lasts 14 h to multiple short sessions. After that, close sessions (e.g., sessions 2 and 3) are merged together.

Table 3. Comparison of result with traditional referrer-based method, intermediate and final result of time-referrer-based heuristics.

No.	Time	URL	Referrer	Traditional Referrer-Based	Intermediate Result of Proposed Method	Final Result of Proposed Method
1	2015-12-30 12:00:00	A	-	1	1	1
2	2015-12-30 12:01:00	B	A	1	1	1
3	2015-12-30 12:03:00	C	B	1	1	1
4	2015-12-30 12:05:00	D	B	1	1	1
5	2015-12-30 12:53:00	E	F	2	2	2
6	2015-12-30 12:55:00	G	E	2	2	2
7	2015-12-30 13:06:00	H	D	1	3	2
8	2015-12-30 13:43:00	A	-	3	4	3
9	2015-12-30 13:45:00	B	-	4	5	3
10	2015-12-31 02:06:00	I	D	1	6	4

4. Implementation and Workflow

4.1. Implementation

The system consists of four components: log importer, database, log processer, and visualization tool (Figure 2). Log importer is used to parse, clean (only detecting part of the crawler), and import raw HTTP and FTP logs into the database. The log processer is primarily responsible for user identification, crawler detection, and session identification. The database used in the system is built upon an open source database solution, called Elasticsearch. Although it is a relatively new database technology, it provides a distributed, scalable, full-text search engine with a HTTP web interface and supports schema-free JSON documents. Another reason that we select Elasticsearch is that it comes with an open source data visualization plugin called Kibana. Kibana saves lots of development efforts by providing visualization dashboard capabilities on top of the content indexed on Elasticsearch cluster. Because of our particular needs, we also developed a visualization tool for session structure tree on Kibana. In our experiment, it took ~20 min to process one month of log files (6 cores, 12G memory, and Win 7 OS). This is acceptable when dealing with small data, we plan to leverage cluster and cloud computing [27] to speed up the process for large log data (e.g., 10 years of log file).

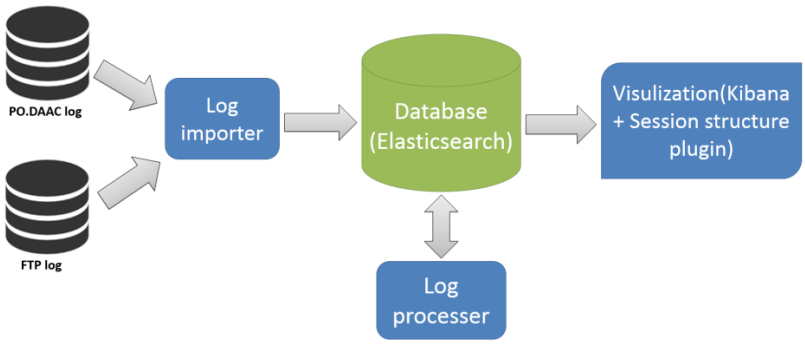


Figure 2. Session reconstruction system architecture.

4.2. Workflow

Overall, the session reconstruction includes seven steps:

- Import HTTP logs: The first step is to import HTTP logs of PO.DAAC website into Elasticsearch. All the redundant requests (.img, .js, *etc.*) and part of the crawler requests are removed based on the known crawler list. Only HTML requests are parsed and imported into database for further processing. The input is 4, 191, 741 raw HTTP logs, and the output is 297, 569 HTML requests in JSON format.

- Import FTP logs: Since there is no user-agent information which is used to compare with crawler list, all the FTP logs (3, 174, 458 logs) are imported into Elasticsearch.
- Synchronize HTTP and FTP logs: Although the combination of user-agent and IP address is preferable, unique user is identified only through IP address since there is no user-agent in FTP log. IPs with maximum sustained request rate greater than two requests are removed from the database. After this step, we found 7536 unique users with 901, 945 logs.
- Time threshold selection: After user identification, we plot the inter-activity histogram based on what we described in the methodology part. Because the expected value of the second curve is several days, we left it out and only focus on the first normal distribution curve. After calculation, we found that the critical value at 97.5% confidence level is around 10 min (596.73 s) (Figure 3).
- Session identification: Both session identification methods are experimented in this step. 15,783 user sessions are found. Based on this result, we further filter session by using the number of request types. Specifically, the numbers of searching, viewing, and downloading requests are required to be no less than 1. When one of the requests is missing (less than 1), the session will not provide valid knowledge as needed for data discovery. In this way, the actual user session that only contains one or two of them, and the remaining sessions that were generated by crawlers, are finally removed. In the end, 414 sessions are identified after this step.
- Similarly, 34,604 user sessions are identified with time-clustering-based heuristics when T is set 30 min, and they are narrowed down to 471 user sessions that contain all three types of requests.
- Structure reconstruction: The last step is to reconstruct the session based on referrer. Note that FTP logs are attached to the nearest viewing request in this process.

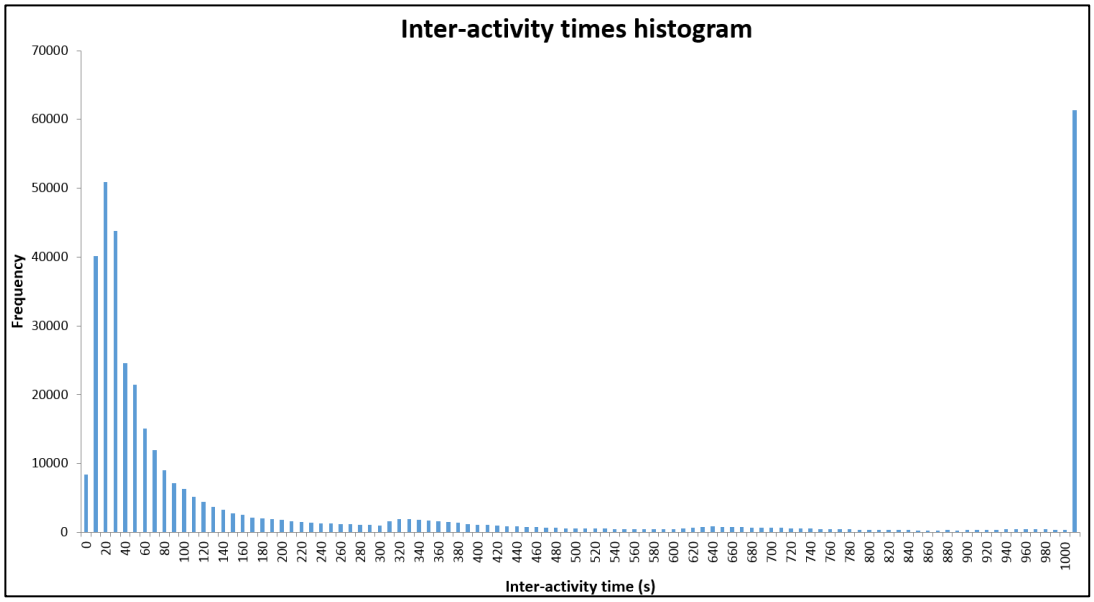


Figure 3. Histogram of inter-activity times.

5. Results

5.1. Comparison of Session Identification Heuristics

Table 4 shows an example from these two session identification results. While the result of time-referrer-based heuristics shows all these sequential requests as a single session, time-clustering-based heurstics splits them into two sessions. Both of them agree that the first few logs belong to the same session, since the sequential time interval between them are quite short (all of them happened in 2 min). However, time-clustering-based heuristics fails at the last second log, because the time interval is larger than 30 min. For time-referrer-based heuristics, because the referrer of the

second last request is identical to the first request and the time interval is also reasonable, they should be assigned to the same session by human interpretation. Although they could be drawn into the same session by setting a large scale in time-clustering-based heuristics, some small sessions will be overlooked since the split process will stop earlier. For example, if we set the time threshold to be 40 min, sessions with length between 30 to 40 min would be difficult to identify. Since the referrer is a valuable information to track user's searching behavior, the time-referred-based heuristics outperforms the time-clustering one.

Table 4. Comparison of time-referrer-based and time-clusteirng-based heuristics.

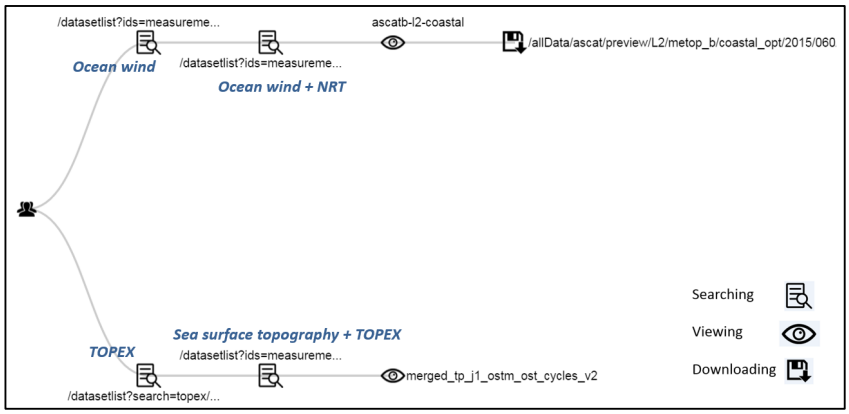
Time	Request	Referrer	Type	Time-Clustering	Time-Referrer
07:19:33	/ghrsst/	-	HTTP	1	1
07:20:04	/datasetlist?search=ghrsst	/ghrsst/	HTTP	1	1
07:20:30	/datasetlist?ids=processinglevel&values=*4*&search=ghrsst&view=list	/datasetlist?search=GHRSST	HTTP	1	1
07:20:34	/datasetlist?ids=processinglevel&values=*3*&search=ghrsst&view=list	/datasetlist?search=GHRSST	HTTP	1	1
07:20:52	/dataset/jpl_ourocean-l4uhfnd-glob-g1sst?ids=processinglevel&values=*4*&search=ghrsst	/datasetlist?ids=ProcessingLevel&values=*4*&search=GHRSST&view=list	HTTP	1	1
07:20:21	/allData/aquarius/L3/mapped/V3/annual/SCI	-	FTP	1	1
07:51:43	/avhrr-pathfinder	/ghrsst/	HTTP	2	1
07:57:00	/seasurfacetemperature	/AVHRR-Pathfinder	HTTP	2	1

5.2. Session Structure

Based on the workflow described, we reconstruct the structure for each session. Figure 4a shows an example from the set of 414 sessions. The user performed two tasks in this session, each corresponding to a branch in the structure tree. This session structure would play a significant role in the next pattern analysis. Both the keywords and the distance between them are critical information to build knowledge base. In addition, Table 5 shows the keywords searched in several sample sessions. The first column shows the session number. Just by visual interpretation, "ghrsst" and "pathfinder" (sessions 3 and 4), and "qscat" and "ascat" (sessions 1, 5 and 12) are more frequently searched than other keyword pairs. In addition, query set that consists of user searching, viewing and downloading behavior could be extracted based on the session structure tree. Figure 4b shows an example from thousands of queries. In this example, the user searched for "quickscat", viewed and downloaded the same dataset "qscat_level_2b_owv_comp_12". This knowledge is helpful to establish relationships between query and dataset and can be integrated for improving data discovery.

Table 5. Keywords searched in sample sessions.

Session ID	Keyword 1	Keyword 2	Keyword 3	Keyword 4
1	qscat	Ascat		
2	pathfinder	Modis	ostia	
3	ghrsst	Pathfinder		
4	pathfinder	Ghrsst		
5	quickscat	Qscat	rapidscat	ascat
6	salinity	aquarius project		
7	geos-3	topex/poseidon	jason-1	
8	long	Ascat		
9	sea level	wind data	climatology sst	sst
10	orumieh	aquarius project		
11	ocean wind	wind speed	quikscat	
12	Quikscat	Ascat		



(**a**)


```
{
  "query": "quikscat",
  "viewdataset": "qscat_level_2b_owv_comp_12",
  "downloaddataset": "qscat_level_2b_owv_comp_12",
  "sessionId": 125.47.246.171@4
}
```


(**b**)

Figure 4. Example session structure results. (**a**) Session structure tree; (**b**) Query set.

5.3. Session Length Histogram and Keywords Popularity

By observing the histogram of session length, we found that most sessions are within one hour, and the maximum length is around five hours (Figure 5). Also, the popularity of keywords is summarized from the final session result. The top 10 keywords are "grace", "ocean currents", *etc.* (Figure 6).

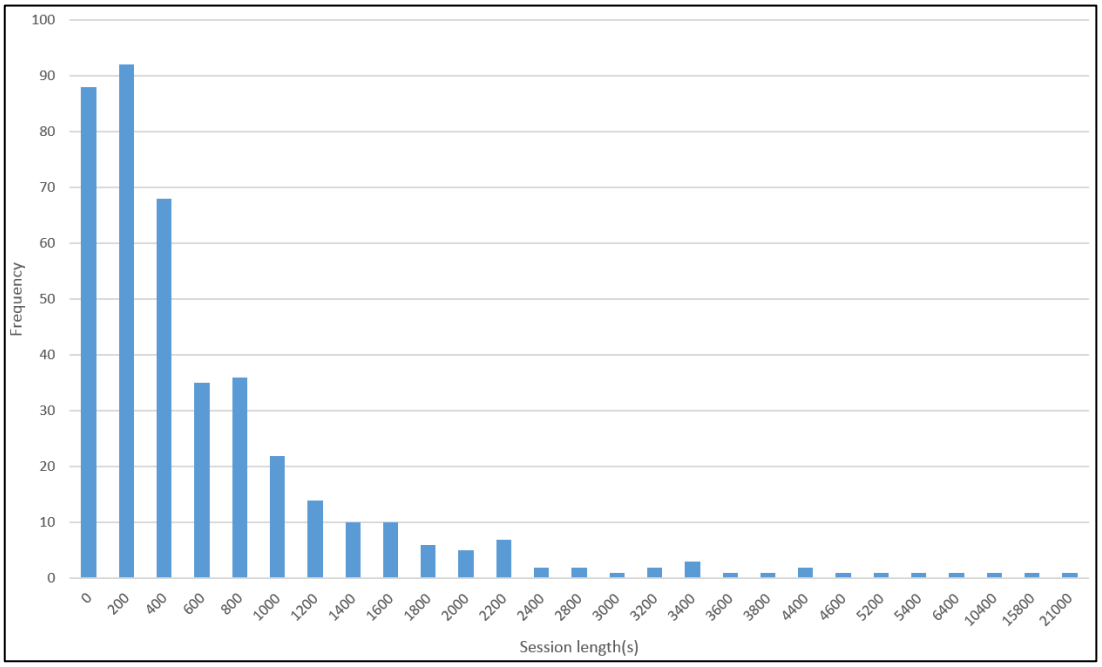


Figure 5. Histogram of session length.

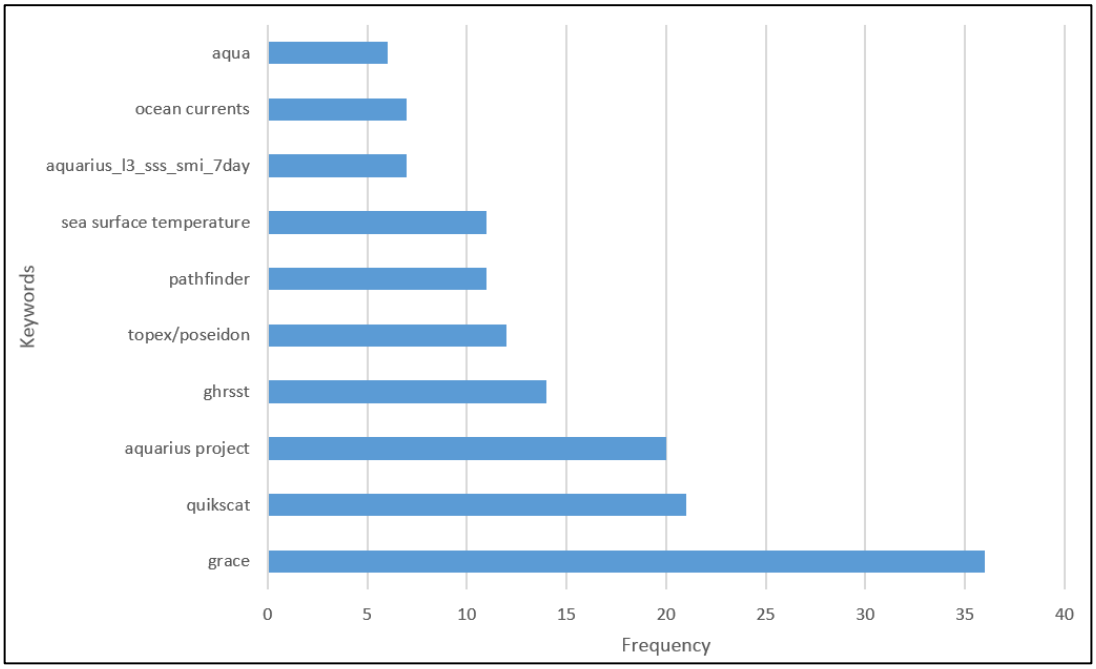


Figure 6. Popularity of keywords.

5.4. Website Traffic and User Location (Heatmap)

People from different places may be interested in different topics and these topics may also change over time. Website traffic and users' location are helpful to build spatiotemporal information into the knowledge base. To do that, IP addresses are converted to geographic locations, and visualized with OpenStreetMap. As can be seen from Figure 7, users come from almost all over the world. There are three clusters in the U.S., one from California, and the other two from the northeastern U.S. To visualize the website traffic, a line chart is created to show how the number of user sessions (not users) changes over time (UTC) (Figure 8). It is a bit surprising that a daily pattern is not found in the chart. A possible explanation is that users come from many different time zones. Further analysis needs to be done here.

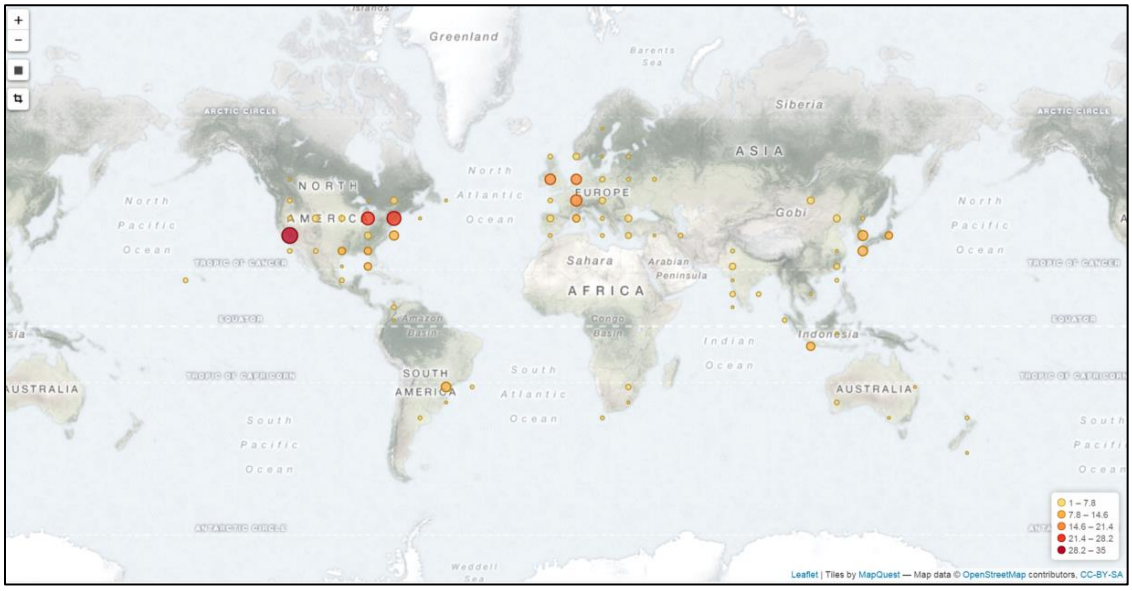


Figure 7. Heat Map of where users come from.

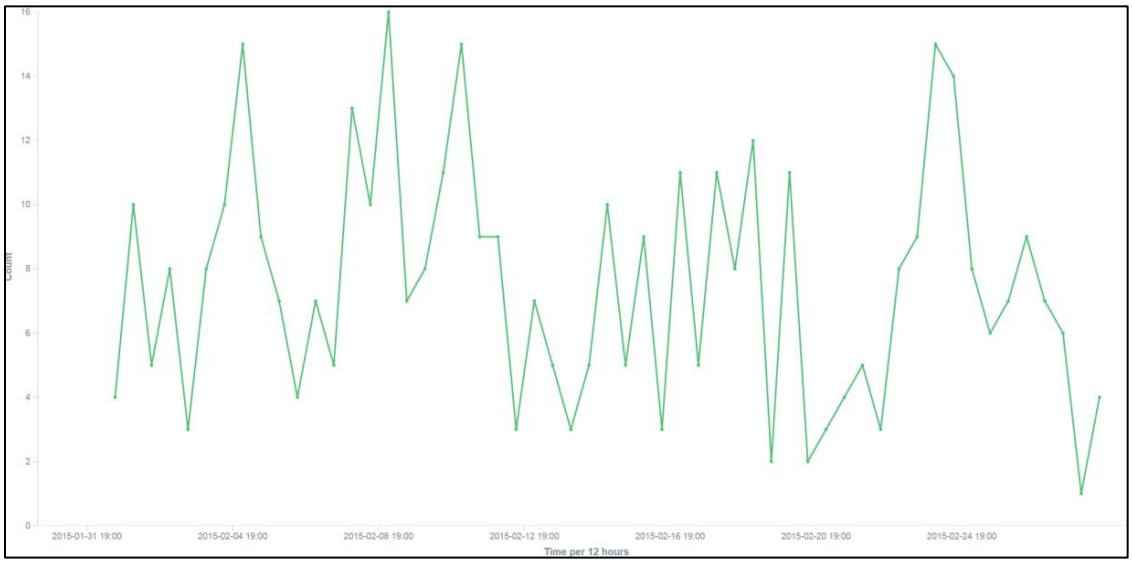


Figure 8. Website traffic.

6. Discussion and Conclusions

This article reports our research on session reconstruction from raw web logs and proposes two session identification methods including time-clustering-based and time-referrer-based methods.

- The proposed selection method based on inter-activity statistical threshold provides more confidence for further analysis in contrast to the empirical time threshold.
- In comparison to the standard referrer-based heuristic, the time-referrer-based heuristic improves the performance from two aspects by introducing a time component: First, close referrer-based tasks are connected to form an actual session, which means the connections among these close tasks are kept this way. Second, a time component adds a dynamic time frame as a restriction to the searching of the previous page, which avoids the generation of an unreasonably long session.
- When compared with the standard time-based heuristics, clustering-based heuristic addresses the limitation of a fixed threshold by building a hierarchy of clusters on the time dimension.
- The workflow of session reconstruction from multiple servers has proven to be able to extract and visualize valuable information from raw log data, which has laid the foundation of discovering keyword and dataset relationships. Furthermore, this information is easy to generalize and reuse in other web usage mining research.

There are several directions where this research could be further improved. In terms of session reconstruction itself, we plan to conduct accuracy assessment by using web logs that contain session identifier information. Although a time-referrer-based heuristic is chosen in our case since the referrer is an important piece of information to tracking user's searching behavior, we believe that the time-clustering heuristic outperforms a traditional time-based heuristic with a fixed threshold, especially for frame-free sites where referrer information is not so important. Quantitative accuracy comparison between time-referrer-based heuristic and time-clustering will be made under different site structures, *i.e.*, frame-base and frame-free sites [8]. Horizontally, the workflow and session identification methods can be applied to other data systems such as polar and biological data. Knowledge extracted from different domains could be integrated to build a more robust and comprehensive knowledge base. Based on the keyword pairs extracted, we will vertically integrate semantic search by building a structured knowledge base [28]. Specifically, further efforts need to be made to discover latent semantic relationships among various queries. For example, we can calculate the probability and identify rules for which two keywords are searched for along with each other.

Unlike traditional association rules mining, the distances of different queries in the session structure tree provide us with a unique advantage to better explore their relationships. Potential techniques that could be leveraged include association rule, sequence learning, and Markov chains. Additionally, we plan to integrate the analysis results of users' query history, clicking behavior, metadata, and existing ontology to augment user queries by traversing the integrated knowledge base, and reveal the actual intent of user searches. Cloud computing will be utilized to facilitate the computationally intensive mining process [27]. Once the semantic knowledge base is successfully constructed, the ultimate goal is to improve the data discovery with better ranked results, related data recommendation, and ontology navigation [29].

Acknowledgments: This project is funded by NASA AIST (NNX15AM85G) and NSF (IIP-1338925). Kai Liu provided valuable information on semantic search. The research was partially carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

Author Contributions: Chaowei Yang came up with the original research idea. Yongyao Jiang developed threshold selection method and clustering-based heuristics; Yun Li developed time-referrer-based heuristics; Yongyao Jiang, Yun Li, and Chaowei Yang developed the workflow and implemented the data processing system; Edward M. Armstrong provided valuable advice to threshold selection and session identification; Thomas Huang and David Moroni provided substantial feedback to the system development; Edward M. Armstrong and David Moroni evaluated the results; Yongyao Jiang, Yun Li, Chaowei Yang wrote the paper; Edward M. Armstrong, Thomas Huang, and David Moroni revised the paper.

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

References

1. Vatsavai, R.R.; Ganguly, A.; Chandola, V.; Stefanidis, A.; Klasky, S.; Shekhar, S. Spatiotemporal data mining in the era of big spatial data: Algorithms and applications. In Proceedings of the 1st ACM SIGSPATIAL International Workshop on Analytics for Big Geospatial Data, Redondo Beach, CA, USA, 7–9 November 2012.
2. Gui, Z.; Yang, C.; Xia, J.; Liu, K.; Xu, C.; Li, J.; Lostritto, P. A performance, semantic and service quality-enhanced distributed search engine for improving geospatial resource discovery. *Int. J. Geograph. Inf. Sci.* **2013**, *27*, 1109–1132. [CrossRef]
3. Yang, C.; Sun, M.; Liu, K.; Huang, Q.; Li, Z.; Gui, Z.; Jiang, Y.; Xia, J.; Yu, M.; Xu, C.; Lostritto, P. Contemporary computing technologies for processing big spatiotemporal data. In *Space-Time Integration in Geography and GIScience*; Springer Netherlands, 2015; pp. 327–351.
4. Langille, A.N.; Meyer, C.D. Google's PageRank and Beyond: The Science of Search Engine Rankings. Available online: http://geza.kzoo.edu/~erdi/patent/langvillebook.pdf (accessed on 3 January 2016).
5. Lei, Y.; Uren, V.; Motta, E. Semsearch: A Search Engine for the Semantic Web. Available online: http://kmi.open.ac.uk/publications/pdf/semsearch_paper.pdf (accessed on 3 January 2016).
6. Srivastava, J.; Cooley, R.; Deshpande, M.; Tan, P.-N. Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data. Available online: http://nlp.uned.es/WebMining/Tema5.Uso/srivastava2000.pdf (accessed on 3 January 2016).
7. Romero, C.; Espejo, P.G.; Zafra, A.; Romeroand, J.R.; Ventura, S. Web usage mining for predicting final marks of students that use Moodle courses. *Comput. Appl. Eng. Educ.* **2013**, *21*, 135–146. [CrossRef]
8. Berendt, B.; Mobasher, B.; Nakagawa, M.; Spiliopoulou, M. The impact of site structure and user environment on session reconstruction in web usage analysis. In *WEBKDD 2002-Mining Web Data for Discovering Usage Patterns and Profiles*; Springer: Berlin, Germany, 2003; pp. 159–179.
9. Zhang, J.; Ghorbani, A. The reconstruction of user sessions from a server log using improved time-oriented heuristics. In Proceedings of the Second Annual Conference on Communication Networks and Services Research, Fredericton, NB, Canada, 19–21 May 2004.
10. Sharma, N.; Makhija, P. Web Usage Mining: A Novel Approach for Web User Session Construction. *Glob. J. Comput. Sci. Technol.* **2015**, *15*, 23–27.
11. Jones, R.; Klinkner, K.L. Beyond the session timeout: Automatic hierarchical segmentation of search topics in query logs. In Proceedings of the 17th ACM Conference on Information and Knowledge Management, Napa Valley, CA, USA, 26–30 October 2008.

12. Cooley, R.; Mobasher, B.; Srivastava, J. Data preparation for mining world wide web browsing patterns. *Knowl. Inf. Syst.* **1999**, *1*, 5–32. [CrossRef]
13. Halfaker, A.; Keyes, O.; Kluver, D.; Thebault-Spieker, J.; Nguyen, T.; Shores, K.; Uduwage, A.; Warncke-Wang, M. User session identification based on strong regularities in inter-activity time. In Proceedings of the 24th International Conference on World Wide Web, Florence, Italy, 18–22 May 2015.
14. Pei, J.; Han, J.; Mortazavi-asl, B.; Zhu, H. Mining Access Patterns Efficiently from Web Logs. Available online: http://www.cse.msu.edu/~cse960/Papers/usagemining/pei00mining.pdf (accessed on 3 January 2016).
15. Zaiane, O.R.; Xin, M.; Han, J. Discovering web access patterns and trends by applying OLAP and data mining technology on web logs. In Proceedings of the IEEE International Forum on Research and Technology Advances in Digital Libraries, ADL 98, Santa Barbara, CA, USA, 22–24 April 1998.
16. Spiliopoulou, M.; Mobasher, B.; Berendt, B.; Nakagawa, M. A framework for the evaluation of session reconstruction heuristics in web-usage analysis. *Inf. J. Comput.* **2003**, *15*, 171–190. [CrossRef]
17. Liu, B. *Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data*; Springer Science & Business Media: Berlin, Germany, 2007.
18. Tanasa, D.; Trousse, B. Advanced data preprocessing for intersites web usage mining. *IEEE Intell. Syst.* **2004**, *19*, 59–65. [CrossRef]
19. Tanasa, D. Web Usage Mining: Contributions to Intersites Logs Preprocessing and Sequential Pattern Extraction with Low Support. Available online: https://tel.archives-ouvertes.fr/tel-00178870/document (accessed on 1 January 2016).
20. Apache. Apache HTTP Server Version 2.4. Available online: http://httpd.apache.org/docs/current/logs.html#combined (accessed on 1 January 2016).
21. Castaglia. ProFTPD Server Logfile. 2009. Available online: http://www.castaglia.org/proftpd/doc/xferlog.html (accessed on 1 January 2016).
22. Romero, C.; Ventura, S.; García, E. Data mining in course management systems: Moodle case study and tutorial. *Comput. Educ.* **2008**, *51*, 368–384. [CrossRef]
23. Doran, D.; Gokhale, S.S. Web robot detection techniques: Overview and limitations. *Data Min. Knowl. Discov.* **2011**, *22*, 183–210. [CrossRef]
24. Benaglia, T.; Chauveau, D.; Hunter, D.R.; Young, D.S. Mixtools: An R package for analyzing finite mixture models. *J. Stat. Softw.* **2009**, *32*, 1–29. [CrossRef]
25. Jenks, G.F. The data model concept in statistical mapping. *Int. Yearb. Cartogr.* **1967**, *7*, 186–190.
26. ESRI. What Is the JENKS Optimization Method? 2012. Available online: http://support.esri.com/en/knowledgebase/techarticles/detail/26442 (accessed on 3 January 2016).
27. Yang, C.; Xu, Y.; Nebert, D. Redefining the possibility of digital Earth and geosciences with spatial cloud computing. *Int. J. Digit. Earth* **2013**, *6*, 297–312. [CrossRef]
28. Liu, K.; Yang, C.; Li, W.; Gui, Z.; Xu, C.; Xia, J. Using semantic search and knowledge reasoning to improve the discovery of earth science records: An example with the ESIP semantic testbed. *Int. J. Appl. Geos. Res.* **2014**, *5*, 44–58. [CrossRef]
29. Yang, C.; Li, W.; Xie, J.; Zhou, B. Distributed geospatial information processing: Sharing distributed geospatial resources to support Digital Earth. *Int. J. Digit. Earth* **2008**, *1*, 259–278. [CrossRef]