Evaluation of the Demand for Online Scientific Publication by Web-Analytics Tools

Yurii Revyakin[0000-0002-3655-6217]

Keldysh Institute of Applied Mathematics, Miusskaya sq., 4, Moscow, 125047, Russia
revyakin@keldysh.ru

Abstract. The work is devoted to the use of web-analytics methods to evaluate the usage of online scientific publications. The practical issues of creating a software tool for obtaining such an evaluation based on log data analysis are considered. The problem of recognition of requests from web robots to correctly determine the indicators of attendance of scientific resources on the Internet is discussed.

Keywords: online publication, web-analytics, log-file analysis, web-robots, request recognition

1 Web analytics: history, subject matter and basic concepts

1.1 The emergence of web analytics and the subject of study

As the World Wide Web evolved from a research project into a global distributed information network, there was an immediate need for a tool that would enable it:

- to estimate the network users' interest in the materials published in the WWW;
- to understand the extent to which the chosen form of presentation corresponds to the tasks set out in their publication.

The answer to this need was the emergence of web-analytics as a set of methods for collecting and analyzing quantitative and qualitative data on the use of Internet resources.

According to what has already become a common definition [1], web-analytics is understood as the process of collecting, measuring, presenting and analyzing data on attendance of an Internet resource to assess and optimize the use of the resource by network users.

1.2 Basic concepts of web analytics

The main result of using web-analytics tools is the generation of reports reflecting the relationship between two types of data: metrics and parameters (dimensions).

Metrics are quantitative indicators of the use of the Internet resource. A metric can be a counter, an average value or a ratio of two metrics.

But metrics are only quantitative indicators of the use of an internet resource and do not represent independent value. Therefore, in web analytics they are considered in relation to the set of attributes that characterize a visitor to a website and their activity during a visit to a web resource. Such attributes are called parameters (dimensions). Parameter values, in contrast to metrics, are text values.

Thus, web analytics reports represents two-dimensional tables where the rows correspond to possible parameter values and the columns contain the metric values calculated for these values.

2 The task of evaluating the demand for a scientific online publication

As the World Wide Web has evolved into a global platform for the exchange of scientific information and the displacement of traditional "paper" forms of publishing scientific articles, interest in evaluating the demand for online publications is no longer determined solely by the legitimate curiosity and vanity of the authors of the publications themselves. The evaluation of the demand for online publications has begun to be used as the most important scientometrics indicator of the relevance and level of research being conducted, to be taken into account when allocating funds to libraries for subscription to paid electronic resources and when determining the rating of the electronic publications themselves.

Therefore, it is not surprising that the discussion on how and based on what indicators must be determined the demand for online scientific publications has been actively pursued over the past 10 years and has not lost its relevance today.

The approach in which the demand for a scientific publication is directly linked to its citation rate is quite widespread. The reliability of this assessment is largely based on the assumption that the researcher who made the reference to the scientific article has read it and considers the information presented in it to be quite interesting. But as noted in [2], this is not always the case for a number of quite objective reasons.

Another alternative approach is to use statistical data describing various aspects of access to the electronic resource to evaluate the demand for electronic publications. This can be the number of requests to a digital library catalogue, the number of downloads of the full text of a publication, or the time spent by a user in studying a publication online. Such statistics provide more detailed information on the use of the digital resource by users, although, unlike the first approach, it is not possible to determine how readers treat the content of a publication. Nevertheless, despite the usual skepticism about the accuracy of any statistics, the idea of creating a single mechanism for collecting and sharing statistical data on the usage of electronic publications has received sufficient support in the professional environment of librarians, publishers and owners of electronic resources. This is confirmed by a series of standards developed by the international non-profit organization COUNTER [3].

The fact that the vast majority of modern digital libraries and repositories are implemented as web services makes it much easier to obtain and process statistical data on the use of these resources. Some of the techniques and criteria of web analytics can be

directly used to evaluate the effectiveness of their use, while others require adaptation and rethinking [4].

3 Selection of criteria for evaluating the demand for a scientific publication

A fairly clear indicator of the demand for a scientific online publication is the number of requests for the full text of the publication itself. Such statistics unambiguously characterize the network users' interest in the material presented and can be included in a simple and concise form directly in the description of the publication in the electronic catalogue of publications. From the point of view of web analytics, this indicator is a metric that can be easily calculated based on the analysis of users' requests for an internet resource.

4 Log file of the web server as a source of raw data for assessing demand for the publication

When a user views an internet resource, each request to upload an html page and its components is usually recorded by the web server in a separate log file. The format of the log file entry is determined by the relevant web server configuration directive, and this entry contains sufficient details about the request and its source.

Here is an example of a log file entry that remains after the web server has processed the request to the main page of the website www.keldysh.ru:

> 194.226.57.126 - - [11/May/2020:16:48:49 +0300] "GET / HTTP/1.1" 200 16541 "-" "Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/60.0.3112.90 Safari/537.36"

The fields of the submitted record contain all the main attributes of the request executed by the web server: source IP address; time stamp of the request execution; query command line, including the URL of the requested resource; description line of the remote client application.

Thus, having access to the log file of the web server, the task of determining the number of visits to the full texts of scientific publications is quite simple and does not require further discussion.

5 "Artificial" Internet traffic

But in order to obtain a reliable estimation of the demand for an online publication, it is necessary to be able to distinguish and exclude requests generated by network applications that automatically scan the pages of an online resource. Such network applica-

tions are commonly referred to as web robots. Preliminary recognition of requests coming from web robots is also relevant because scientific publications are primarily of interest to a fairly small number of specialists and the number of their potential readers is relatively small. Practice shows that the number of queries per day to the text of a scientific publication from real users rarely exceeds two or three dozen; at the same time, according to statistics given in [6], about one third of all queries addressed to digital libraries come from robots.

Before we discuss methods for identifying requests coming from web robots, we will try to figure out who uses them and why. Web robot programs are primarily used by all major Internet search engines (Google, Yandex, etc.) to extract information from web pages and then index it in their databases. Such programs are subject to certain agreements when scanning websites and explicitly mark themselves in the header of the website query. In this case, it is no more difficult to determine the requests coming from web robots. There are many more such specialized network applications used to collect commercial and social information on the internet. The extent to which they meet generally accepted network standards in their implementation depends entirely on the goodwill of their developers. But web robot programs can also be used as malicious network applications. One of the most common scenarios for using them in this capacity is an attack on a website server with multiple requests, the aim being to cause the server to refuse to serve further requests (the so-called DDoS attack). A more sophisticated example of malicious use of robot software is the technology for sending out unwanted advertising information called "referrer spam". Web robot are used to automatically generate and send http requests where the URL of the advertised website is intentionally specified as the address of the page which is the source of the request. In this way, the URL of the advertised site is not only included in reports generated by the web analytics system, but also indexed by search engine robots, which increases the search rating of the advertised site. As a rule, malicious web robots ignore the rules limiting the scanning of documents for a web resource and disguise their activity by falsifying the name of the program that sends the request in the header of the http-request. To detect such requests in the log file, more complex approaches and software algorithms have to be used.

It should be noted that the victims of web robots on the Internet are not just online libraries of scientific publications. Since the volume of traffic entering the site has turned into a commodity with its quite specific value, there have been many technologies to manipulate this value. And to implement these technologies, more and more new web robot programs are being created, increasing the already significant share of "artificial" traffic on the Internet.

6 Methods for recognizing requests from web robots

If the server log file data is used as a source of information to determine the popularity of a scientific publication, then the determination of requests coming from web robots is performed at the stage of initial parsing the log file records. The methods used for recognition are very diverse and can range from simple verification of the query fields

to match a list of predefined strings to complex analytical algorithms using probability models and machine learning. The classification of such techniques is discussed in detail in [5].

The most obvious way of detecting robots' queries consists of simply comparing individual fields of a log file entry with predefined lists of key values. In this way, we can detect a request from a robot program by the identifier of the program application specified in the description line of the request source. Or we can compare the IP address of the request source with a database of addresses actively used by the web robots. This method of identifying requests from robots is simple and unambiguous, but its application is based on two significant assumptions:

- query fields contain valid information;
- a set of key values that can be used to define a web robot by the content of the log file fields is known in advance.

It should also be noted that invalid values for some log file entry fields may also be considered as a sign that the request belongs to a web robot. For example, "anonymous" requests with an undefined field describing the source of the request mainly come from web robots programs that violate network protocols and agreements.

Other methods of detecting requests from robots are based on an analysis of the general characteristics of a sequence of requests related to a single user visit. Such characteristics may include: the distribution of queries according to the methods used and the types of web resources requested, the intensity of queries per unit of time and the amount of data transferred. The procedure for separating requests from robots is broken down into two stages: first, the necessary metrics are calculated for all requests from one visit and then their values are used to classify the source of the request. This classification is based on criteria that are most often determined empirically: it is known, for example, that web robots programs, while browsing the site, more often request only document attributes and, as a rule, do not upload image files placed on the web page. The final decision on the nature of the source of the requests is made provided that several such criteria are met at the same time. It must be emphasized that the decision taken is only correct for the sequence of requests reviewed, which relate to the same user visit. The mechanism of dynamic IP addresses allocation does not make it possible to extend the selection made to all requests originating from a given IP address.

The development of the previous group's methods are algorithms based on machine learning and the application of probability models. As in the previous approach, the first step is to identify sequences of queries related to a single user visit, and a set of characteristics is calculated for each individual visit. Then, using a pre-constructed probabilistic model, a decision is made on the nature of the source of the requests for each visit (a real visitor or a robot program). We will not dwell on the methods of this group in detail, as they are not used in the implementation of counter of requests for the full text of the online publication, which will be discussed below.

7 Counter of the number of requests for full text publications of the KIAM RAS online library

The KIAM RAS online library has its own tool for calculating the number of accesses to full texts of publications placed in the library, implemented as a separate software utility. Both methods using a syntactic analysis of the query fields and methods based on an analysis of the characteristics of user visits are used to determine the nature of the source of the query. The program does not contain any tools for generating statistical reports, but only transfers the calculated parameters for further processing to the online library software. After processing, data on the number of requests for the full text of the publication is submitted directly to the card of the publication itself in the online library catalogue. The program is implemented in Ruby language [10] and uses additional libraries to parse the web server log file entries, access databases and external data sources in JSON format.

8 Request counter – description of the algorithm

It is possible to break down the algorithm to calculate the number of requests for full texts of publications into three main stages:

- analysis of primary data and highlighting requests for full text of online publications. The primary data source is the Apache web server log file version 2.2;
- definition of requests coming from web robot's programs;
- counting the number of readers' requests for complete full text of online publications and recording the results in the online library's service database.

8.1 Raw data processing

A database (DB) of queries to the web server is being created. All records contained in the web server log file are sequentially read out and for each entry of the log file a database record is created containing the following information about the initial http request:

- the IP address of the request;
- type of http-request;
- query creation time;
- request status code;
- URL of the requested resource;
- the description line of the client application that created the request, including the identification of the client program.

The MongoDB DBMS [9] is used for storing information about the queries being processed. MongoDB is a document-oriented, non-relational DBMS that allows storing documents of different structures in one database. Programming of queries to objects stored in the database is largely facilitated by the use of Mongoid – a toolkit for the

Ruby language, which implements an object-oriented interface for access to the MongoDB DBMS.

8.2 Primary data analysis

Successful http requests to files containing full texts of publications are selected among all the requests entered in the database. It is not difficult to select such requests, as the names of files with full text of online publications are formed in a uniform manner, in accordance with agreements adopted in the KIAM RAS online library.

8.3 Identification of requests coming from web robots programs

The recognition of requests from robots is carried out in several stages, successively applying a number of program tests, each of which uses its own criteria for selecting requests. Let us look at the methods used to test queries in the current version of the program in more detail.

Identification of the source of requests by name of the software application. Records of requests for files with full text publications are checked to ensure that the name of the application - the source of the request - matches the list of application names used by web robot programs. Application names of web robot are stored in a separate database, which is formed on the basis of a freely distributed list of regular expressions presented in JSON format [7]. Each regular expression determines all variations in the inclusion and spelling of the robot program name in the source line of the http request. The list contains over 400 regular expressions corresponding to the identifiers of the most common web robot programs and is updated periodically. Requests that are defined as coming from a robot by the name of the client program are excluded from further consideration.

Checking the source IP address of the request. For each request the source IP address is checked for belonging to the subsets of addresses used by web robots. The address ranges are stored in a separate database generated from the open source text list [8].

Checking to request a fragment of the file. The http-protocol specification makes it possible to generate a request to transfer only a part of a document by specifying in the request header the byte range of the requested fragment. In this way, it is possible to break down the loading of a large file into several parts and use your own request to load each part. This property of the http protocol is widely used by modern browsers to optimize and parallelize the upload process. But from the point of view of web-statistics, the sequence of requests for downloading file fragments should be regarded as one user's request to a document. There are no clear formal criteria for distinguishing such a sequence of requests - each browser implements its own algorithm for splitting the loading of a large document into several requests. One solution to this problem is therefore based on the assumption that all requests for full or partial file downloads coming from a specific IP address within a fairly short period of time are the result of a single user request of the document. Of the sequence of queries thus allocated, only the first time query is considered further, while the rest are ignored.

User session identification. The next step in the selection of requests is to consider the general characteristics of the user visit, during which a request for the full text of the publication was made. For this purpose, all requests relating to this visit must be highlighted in advance. It is considered that all queries that meet the following conditions are included in the same user visit together with the current request for the text of the publication:

- sent from the same IP address as the current request;
- created by the same client application (the strings of the request source description are the same);
- the time interval between successive requests in a session does not exceed a predetermined threshold value.

Checking for downloading the robots.txt file during the session. The robots.txt file was developed by the W3C consortium to manage the behaviour of web robots when they bypass the site. This text file, located in the root directory of the site, contains instructions that regulate web robots access to individual files or directories on the site. Following the instructions given in the robots.txt file is voluntary for web robots, but most robots of known search engines stick to them and download the robots.txt file at the beginning of each session of browsing the site. The fact that the robots.txt file is downloaded during a session is therefore very likely to indicate that all requests in the session come from the robot software and can be excluded from further consideration.

8.4 Verified requests from real online library users

All requests that have passed the selection stages described above are regarded as verified requests for full texts of publication from real online library visitors.

8.5 Counting the number of requests and recording the results

By highlighting the requests that have been identified as coming from real readers, it is not difficult to calculate the total number of visitors' requests to the full text of each publication. This data is recorded in the online library's SQL server service database. The data on the number of hits for each printout is represented in the database by records of the following format:

- date of application;
- URL of the full text publication file;
- number of requests to the file.

The online library database server is implemented on the Microsoft SQL Server platform; to access the service database, an object-oriented interface to SQL database, implemented by the Ruby Sequel package, is used.

9 Presentation of statistics on access to online publication

The counter results recorded in the service SQL database are then processed by the online library software and used to update the statistics on requests in the main database

of electronic publications. The aggregate statistics of references to online library publications are presented on the electronic cards of each publication in the following format (see Fig. 1):

- total number of requests from the day of publication of the print or from the day of launching the program for calculating requests for print (09.01.2020);
- number of requests for the last 30 days;
- the difference in the number of requests as compared to the previous 30-day statistical collection interval.

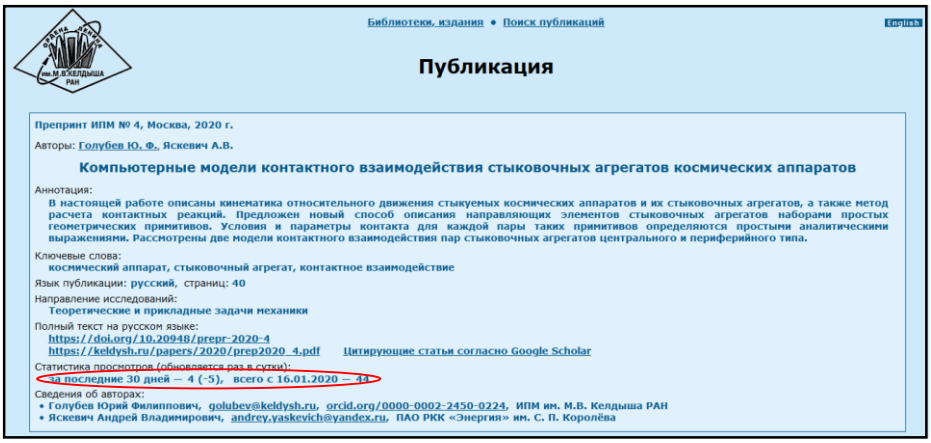

Fig. 1. Online publication card with information on statistics of requests (highlighted in red)

10 Conclusion

The need to evaluate the demand for online scientific publications is becoming more and more urgent as traditional "paper" technologies for the dissemination of scientific knowledge are being replaced by internet technologies that offer qualitatively new opportunities for information exchange. The development of the software tool presented here had two main goals:

- to obtain statistics of requests to the full text of the publication with sufficient reliability;
- to present data on the demand for a publication in a simple and concise form, including them in the metadata of the description of the publication in the online library.

The implemented mechanism for identifying the source of requests allows for the recognition of requests coming from web robots, by successive application of a chain of software tests; the set of such tests can be extended in subsequent versions of the program.

References

1. Booth, D., Jansen, B.J.: A review of methodologies for analyzing websites. Handbook of Research on Web Log Analysis, IGI Global, pp. 143–164 (2009), https://doi.org/10.4018/9781605669823.ch009.
2. Frost, C.O.: The Use of Citations in Literary Research: A Preliminary Classification of Citation Functions. The Library Quarterly: Information, Community, Policy, vol. 49, no. 4, pp. 399-414 (1979).
3. COUNTER project official site. URL: https://www.projectcounter.org/.
4. Reviakin, Iu.G.: Vozmozhnosti web-analitiki dlia otsenki effektivnosti nauchnykh publikatsii. Preprinty IPM im. M.V. Keldysha, vol. 50 (2020), https://doi.org/10.20948/prepr-2020-50.
5. Doran, D., Gokhale, S.S. Web robot detection techniques: overview and limitations. Data Min Knowl Disc 22, pp. 183–210 (2011). https://doi.org/10.1007/s10618-010-0180-z
6. Huntington, P., Nicholas, D., & Jamali, H.R.: Web robot detection *in the* scholarly information environment. Journal of Information Science, 34(5), pp. 726–741 (2008).
7. Syntactic patterns of HTTP user-agents used by bots / robots / crawlers / scrapers / spiders. URL: https://github.com/monperrus/crawler-user-agents.
8. Free IP2Location Firewall List by Search Engine. URL: https://www.ip2location.com/free/robot-whitelist.
9. MongoDB official site. URL: https://www.mongodb.com.
10. Ruby programming language official site. URL: https://www.ruby-lang.org.