Web Graph Analyzer Tool

Konstantin Avrachenkov INRIA Sophia Antipolis 2004, Route des Lucioles, B.P.93 06902, France K.Avrachenkov@sophia.inria.fr Danil Nemirovsky St.Petersburg State University 35, Universitetsky Pr.,Peterhof St. Petersburg, 198504, Russia Danil.Nemirovsky@gmail.com Natalia Osipova INRIA Sophia Antipolis 2004, Route des Lucioles, B.P.93 06902, France Natalia.Osipova@sophia.inria.fr

ABSTRACT

We present the software tool "Web Graph Analyzer". This tool is designed to perform a comprehensive analysis of the Web Graph structure. By Web Graph we mean a graph whose vertices are Web pages and whose edges are hyperlinks. With the help of the Web Graph Analyzer we can study the local graph characteristics such as numbers and sets of incoming/outgoing links to/from a given page, the page level relative to a given root page, and the global graph characteristics such as PageRank, Giant Strongly Connected Component, the number of dangling nodes. The Web Graph Analyzer has a user friendly GUI that allows an easy collection of a part of WWW and its thorough investigation. The Web Graph Analyzer is based on the Oracle DBMS which scales well with the large volumes of data.

Categories and Subject Descriptors

G.2.2 [Discrete Mathematics]: Graph Theory—Graph algorithms; G.4 [Mathematical Software]: Metrics—Algorithms design and analysis, User interfaces; H.3.3 [Information systems]: Information Storage and Retrieval— Information Search and Retrieval

General Terms

Graph Theory, Algorithms

Keywords

World Wide Web (WWW), Web Graph, Software Tool, Crawler, Connectivity, PageRank

1. INTRODUCTION

During the last decade the information network World Wide Web (WWW) has transformed our society and economy. One of the main reasons behind the amazing success of WWW is its hyper-link structure. Hyperlinks are the pointers from some particular places of Web pages to other Web

Valuetools '06, October 11-13, 2006, Pisa, Italy

Copyright 2006 ACM 1-59593-504-5 ...\$5.00.

pages. They are alike citations in scientific literature. However, there are important differences between WWW and the network of scientific citations. It is much more common in WWW that two documents have reciprocal links. Furthermore, the Web pages and hyper-links not only can be created but also can be deleted. Thus, WWW is a much more dynamic information network than the network of scientific citations.

Due to the easy accessibility of the Web documents and the small world effect of WWW (almost any two Web documents are a few clicks from each other), the navigation in WWW is very simple. This fact has largely contributed to the popularity of Web surfing. However, even though Web surfing is easy and fun, the Web has billions of documents and it is often a challenging problem for a user to find relevant and high quality information on a subject in which he/she is interested. Numerous Search Engines try to solve the problem of the information search. The Google search engine is among the most successful search engines if not the most successful one. Its success is attributed primarily to the introduction of link-based ranking criterion PageRank [5] for the search results. PageRank criterion takes into account the structure of the Web Graph. By Web Graph we mean a directed graph whose vertices are Web pages and whose edges are hyper-links.

Curiously enough, a large number of search engine optimization companies has been created for the analysis and optimization of the hyper-link structure. Of course, most search engines and search engine optimization companies are very secretive about their studies of the Web Graph. The analysis of the Web graph is also very interesting not only from the commercial point of view but also its analysis is very interesting and challenging from the academic point of view. Still there are virtually no software tools available for the comprehensive, efficient and user friendly analysis of the Web Graph. Here we present such a software tool which we call "Web Graph Analyzer".

The rest of the paper is organized as follows: In Section II we define the Web Graph and its most important characteristics. In the ensuing Section III we outline the algorithms for the computation of the Web Graph characteristics. Then, in Section IV we present the Web Graph Analyzer tool and explain its design and its GUI. In Section V we describe the performance test for the Web Graph Analyzer. We conclude the paper with Section VI.

2. WEB GRAPH

The World Wide Web hyper-text structure can be repre-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

sented as a directed graph $G = \{V, E\}$, where Web pages are vertices and hyper-links are edges. In short, the graph of the World Wide Web hyper-text structure is called *Web Graph* (WG). Let $V = \{1, 2, ..., |V|\}$ denote the set of Web Pages and let $E = \{v \to u | v, u \in V\}$ denote the set of hyper-links. Thus, the notation $v \to u$ means that there is a hyper-link from page v pointing to page u. If there are multiple hyper-links from one Web page to another, we consider these multiple hyper-links as a single edge of the Web Graph. We also disregard the hyper-links inside a single Web page. Therefore, the Web Graph does not have loops, but, of course, it has cycles.

The Web Graph has local and global characteristics. Among local characteristics the most important are: numbers and sets of incoming/outgoing links to/from a given page, the page level relative to a given root page and also average values and distributions of these characteristics. Among global characteristics we can mention PageRank (PR), Giant Strongly Connected Component (Giant SCC) and the number of dangling nodes.

For completeness of the presentation, let us provide more detailed definitions and properties of the above mentioned characteristics.

Thus, we have the following definition of the number of incoming/outgoing edges.

DEFINITION 1. The number of incoming/outgoing edges to/from a given page is equal to the number of pages which have links pointing/pointed to/by the given page.

DEFINITION 2. The level of a page v relative to a root page u corresponds to the minimal distance from the root page u to the given page v.

In other words, the page level shows what minimal number of "clicks" a user should do to reach the given page starting from the root page. One can extend the above definition to the case of several root pages. In that case the page level is selected as the minimal level among all levels corresponding to the root pages.

DEFINITION 3. A dangling node is a page which does not have links to the other pages of the Web Graph.

PageRank (PR) is the eigenvector centrality measure of the Web Graph. Its elements represent the authority or popularity of Web pages. PageRank is used by the search engine Google as one of the principle criteria to rank answers to a user's query. The formal definition of PageRank is given below.

DEFINITION 4. Denote by n the total number of pages on the Web and define the $n \times n$ hyperlink matrix P as follows. Suppose that page i has k > 0 outgoing links. Then $p_{ij} = 1/k$ if j is one of the outgoing links and $p_{ij} = 0$ otherwise. If a page does not have outgoing links, the probability is spread among all pages of the Web, namely, $p_{ij} = 1/n$. In order to make the hyperlink graph connected, it is assumed that a random surfer goes with some probability to an arbitrary Web page with the uniform distribution. Thus, the PageRank is defined as a stationary distribution of a Markov $chain\ whose\ state\ space\ is\ the\ set\ of\ all\ Web\ pages,\ and\ the\ transition\ matrix\ is$

$$\tilde{P} = cP + (1 - c)(1/n)E,$$
(1)

where E is a matrix whose all entries are equal to one and $c \in (0, 1)$ is the probability of not jumping to a random page (it is chosen by Google to be 0.85). The Google matrix \tilde{P} is stochastic, aperiodic, and irreducible, so there exists a unique row vector π such that

$$\pi \tilde{P} = \pi, \quad \pi \underline{1} = 1, \tag{2}$$

where $\underline{1}$ is a column vector of ones. The row vector π satisfying (2) is called a PageRank vector, or simply PageRank.

PageRank has the following useful interpretation: If a Web surfer follows a hyperlink with probability c and jumps to a random page with probability 1-c, then PageRank π_i can be interpreted as a stationary probability that the surfer is at page i.

DEFINITION 5. The Strongly Connected Component (SCC) is a subgraph of G in which from any page there is a way to any other page of that subgraph. Namely, G^{SCC} is a SCC if $\forall v, u \in G^{SCC} \exists \{v_1, v_2, ... v_n\} \in G^{SCC}$ such that $\exists v \to v_1, v_1 \to v_2, ..., v_n \to u$.

It turns out that in the Web Graph there is a Giant Strongly Connected Component, which is significantly larger than the other SCCs of the Web Graph. For every Web page owner it is important to know, if the page belongs to the Giant SCC or not. In the case when a page does not belong to the Giant SCC the probability for a Web surfer to find it is typically very small.

3. ALGORITHMS DESCRIPTION

Let us outline algorithms that are used in the software tool "Web Graph Analyzer".

3.1 PR computation algorithms

There are several methods for PR computation. The method which is used by Google is Power Iteration (PI) method. It is based on successive multiplications of the PR approximation vector by the transition matrix given in (1).

Algorithm 1. Power Iteration method

- 0. The initial approximation is chosen as the uniform distribution vector $\pi^{(0)} = (1/n)\underline{1}^T$.
- 1. The k-th PR approximation vector is calculated by

$$\pi^{(k)} = \pi^{(k-1)} \tilde{P}, \quad k \ge 1.$$
(3)

2. The method stops when the required precision ε is achieved, i.e., $||\pi^{(k)} - \pi^{(k-1)}||_1 \leq \varepsilon$. If the precision is not yet achieved then return to Step 1.

The number of flops needed for the method to converge is of the order $\frac{\log \varepsilon}{\log c} nnz(P)$, where nnz(P) is the number of non-zero elements of the matrix P [7]. We note that the relative error decreases uniformly for all pages. Several proposals (see extensive survey papers [7] and [3]) have recently been put forward to accelerate the power iteration algorithm.

We emphasize that the implementation of the PI method does not really need to multiply the PR approximation vector by the transition matrix. The transition matrix \tilde{P} is composed of the hyperlinkmatrix P which is very sparse and the rank-one matrix E. The latter fact is explained by the following equivalent form of (3):

$$\pi^{(k)} = c\pi^{(k-1)}P + (1-c)(1/n)\underline{1}, \quad k \ge 1.$$

There is another class of efficient methods for PR computation. These methods are probabilistic Monte Carlo (MC) methods [2, 4]. The Monte Carlo methods are based on simulation of a random surfer going through the Web Graph.

The principle advantages of the probabilistic Monte Carlo type methods over the deterministic methods are: the PageRank of important pages is determined with high accuracy already after the first iteration; MC methods have natural parallel implementation; and MC methods allow continuous update of the PageRank as the structure of the Web changes.

Monte Carlo algorithms are motivated by the following convenient formula that follows directly from the definition of the PageRank:

$$\pi = \frac{1-c}{n} \underline{1}^T [I-cP]^{-1} = \frac{1-c}{n} \underline{1}^T \sum_{k=0}^{\infty} c^k P^k.$$
(4)

There are two interpretations of this formula.

Consider a random walk $\{X_t\}_{t\geq 0}$ that starts from a randomly chosen page. Assume that at each step, the random walk terminates with probability (1-c), and makes a transition according to the matrix P with probability c. It follows from (4) that the end-point of such random walk has a distribution π .

The second interpretation is that π is also a distribution of frequency of visiting all pages during the random walk.

According to these interpretation there are several versions of MC method. In [2] we performed an extensive investigation and comparison of different Monte Carlo methods. Below we present the most efficient version.

Algorithm 2. MC complete path stopping at dangling nodes

For this algorithm we use a slightly different definition of the hyperlink matrix that disregards dangling nodes. Namely, let Q be a modified hyperlink matrix with elements defined as $Q_{ij} = 1/k$, if page i has k > 0 outgoing links, and a link points to page j, and 0, otherwise.

Then, we simulate the random walk $\{Y_t\}_{t\geq 0}$, whose transitions are governed by matrix Q, starting exactly k times from each page. The random walk $\{Y_t\}_{t\geq 0}$ can be terminated at each step either with probability (1-c) or when it reaches a dangling node.

Finally, for any page j, evaluate the estimate of π_j as the total number of visits to page j divided by the total number of visited pages.

3.2 SCC computation algorithm

The next algorithm allows us to determine an SCC, which includes a given root page.

Algorithm 3.

- 0. The root page is selected.
- 1. With the depth search algorithm the set S_1 of all pages reachable from the root page is found.
- 2. With the depth search algorithm the set S_2 of all pages from which the root page is reachable is found.
- 3. $S = S_1 \bigcap S_2$ is an SCC.

3.3 Level computation algorithm

For computing the levels first one or several root pages are selected. The selection is the user's choice. Denote by lvl(u) the level of page u. Then, the level computation algorithm is as follows:

Algorithm 4.

- 0. $S_1 = \{\text{root pages}\}, lvl(u) = 1, u \in S_1$
- 1. $S_2 = \{v \in G \mid \exists u \to v, u \in S_1, lvl(v) = null\}$. If $S_2 = \emptyset$, Stop.
- 2. $lvl(v) = lvl(u) + 1, v \in S_2, u \in S_1.$
- 3. $S_1 = S_2, S_2 = \emptyset$.
- 4. Go to Step 1.

Analyzing the structure of the WG we found that the fraction of pages with high level numbers is small and those pages with high level numbers have very few links. Thus, it seems to be computationally efficient to calculate the PR approximation taking into account only the first few levels of the WG. For such a "restricted" PageRank we have to select the subgraph G^l of the graph G according to the page levels as follows: $G^l = \{u \in G | lvl(u) \leq l\}$.

The notion of levels is also very useful for the computation of query dependent link based criteria [6].

4. WEB GRAPH ANALYZER TOOL

To realize the methods listed in the previous section and to provide a framework to implement the other Web Graph specific methods, we developed a software tool called "Web Graph Analyzer" (WGA).

The WGA tool is designed for the solution of the following problems:

- collecting and storing information about Web pages and hyper-links;
- searching pages/hyper-links by their URL names or by their neighbor pages URLs;
- computing WG local characteristics:
 - finding sets of incoming/outgoing links for a page;
 - finding in- and out- degree distributions;
 - computing page levels;

- ...

- computing WG global characteristics:
 - computing PageRank with PI and MC methods;
 - SCC detection;
 - ...
- user friendly representation of the WG structure analysis.

4.1 Software Choice

Analyzing the Web Graph, one has to store and to process a huge volume of data. An efficient way to store and to process that huge volume of data is to use a Data Base (DB). The number of Web pages is not only huge but also it grows from day to day. Also the quality and the interest of the results of the most algorithms improves with the increased volume of the collection. This is why when creating a system for the analysis of the WG structure, it is very important to make the system scalable from the very beginning.

To retrieve the information about the WG structure, a special program, Crawler, was developed using Java and PL/SQL. More details about the Crawler are given in one of the subsections below.

When realizing and running any algorithms on the Data Base, we need to extract the data from the DB, to copy it into the memory of the computer, to process it in some way and to write the results back to the DB. The best way to realize all the operations with data stored in the DB is to use a Data Base Management System (DBMS). An example of high-performance, scalable and secure DBMS system is the Oracle DBMS. Oracle DBMS can maintain and process a very large amount of data. Its high performance is due to the use of programming language PL/SQL. Oracle DBMS uses stored procedures which run inside the kernel of DBMS in the most efficient order and provide fastest execution.

As Oracle has the same interpreter for all the operating systems there is no need to recompile the code when the user needs to work on another platform.

The multi-user access is already realized in the Oracle DBMS, so we can organize information retrieval and calculation processes at the same time. For instance, it becomes easy to realize parallel PageRank calculations with the MC method on multiprocessor computer working with the same Data Base. Also we can make run several Crawlers on different computers in a local network updating the same Data Base.

All algorithms mentioned in the previous section are written in the PL/SQL language and are realized as stored procedures, which run inside the kernel of DBMS.

The system principle scheme is presented in Figure 1.

Graphical User Interface is realized using Oracle Developer Tool. It is a special development environment to represent the work with the Oracle DB in the user friendly manner.

4.2 Data Base Structure

Our DB system consists of the following tables:

- pages;
- links;



Figure 1: System structure

- auxiliary tables;
- tables to store information for the Crawler.

The table "pages" is used for page indexation. Every page has its own identity index. In auxiliary tables these indices are used instead of the full pages URLs. This minimizes stored volumes of information and decreases processing time. In the table "pages" all information about the page is stored: page index, URL, PR, number of ingoing and outgoing links, page level, and possibly page content.

Table "links" stores the information about hyper-links between Web pages. For storing the link information the method of adjacency list is used [1]. Since the Web Graph is sparse, the method of adjacency list avoids unnecessary storage of many zero elements of the hyperlink matrix. Furthermore, the adjacency list not only saves a lot of storage space but also simplifies significantly the operations with WG.

The auxiliary tables are used to store all the intermediate data which is need to implement the algorithms.

Tables with the information for crawler store such information as the initial page URL, the Web filters, the extensions of the pages we want to collect and so on. All this information is needed to organize the data collection from the Web.

4.3 Crawler

The Crawler is designed to collect samples of WG as fast as possible and at the same time to use efficiently and carefully computation and network resources.

A number of Crawler instances can be run in local area network to use computation resources efficiently. All instances use one DB to store the collected data but different computers to retrieve and to parse Web pages. The Crawlers concurrently distribute among themselves Web pages and then treat these pages autonomously.

A flexible system is used to restrict a sector of the crawled Web. The sector can be limited by URLs, IP addresses or by the country of the Web server location. Two lists of regular expressions are used to restrict the sector by URL. The lists are called "filter" and "fence". If a URL satisfies at least one regular expression from the filter list, that page can be downloaded for the further analysis. If a URL satisfies at least one regular expression from the fence list, the Web page with that URL is not downloaded. A URL has to pass both the filter and the fence to be downloaded. Filtration by IP address is carried out in a similar way. One can specify ranges of IP addresses that can be accessed and ranges of IP addresses that are not allowed to be accessed. The IP addresses are resolved by URL. The table of correspondence country-IPaddress is used to filter by country of the server location.

While working the Crawler maintains two buffers of pages: the buffer of downloaded pages and the buffer of new pages. The first one is used to resolve indices of URL received by parsing. The length of that buffer is limited. When the limit is reached the pages from head of the buffer is removed and then new pages are inserted in the tail.

The buffer of new pages is responsible for pages downloading and distributing among Crawler instances. When the buffer becomes empty the crawler tries to get new pages from DB. If its attempt is successful it starts to work on the retrieved pages. If it fails, the Crawler instance goes into the waiting state. The waiting state of the Crawler instance can be noticed by other crawler instances that are working with the same DB. In this case they free a number of their new pages that will be passed to the waiting Crawler instance.

If only a single Crawler instance is run in the local network the mode of its functioning depends on the buffer length of new pages. If the length is one, then the Crawler goes through a chain of pages until it reaches a page without outgoing links. Then it sends a request to DB for a new URL. This mode closely corresponds to the Deapth-First-Search (DFS) traversal method [1]. If the buffer length were unlimited, then the mode corresponds to Width-First-Search (WFS) traversal method [1]. If the buffer length is finite, then we get some traversal method which is in between DFS and WFS.

4.4 Algorithms Implementation

All algorithms of the system are realized as stored procedures in PL/SQL language.

The use of the Oracle DBMS indexation scheme significantly decreases the computation time for all algorithms. Every table in the DB has one or several indices which makes it possible to work with the information in the most efficient order.

The numbers of incoming and outgoing links are efficiently computed using the table "links".

The table "pages" is often updated according to the new / updated information.

The algorithm of PR computation with the PI method is realized in the following way: first the initial values 1/n is assigned to the PR column. Then, for every iteration $\pi^{(k+1)}$ is computed using the previous value of $\pi^{(k)}$ according to the table "links". The updated value of PR on every step is stored in the table "pages".

PR computation with the MC method is realized according to the Algorithm 2. For the MC method realization the recursive procedure which simulates the Web surfer is run many times. It starts a random walk from each record of the table "pages". With the probability c it goes to the randomly selected pages according to the table "links". With the probability 1 - c it starts a new random walk from the next page in the list of pages. When the simulation is stopped, the PR is computed as the fraction of the number of visits of a particular page to the total number of visits. Then the value of PR is updated in the in table "pages". It is easy to organize parallel PR computation with MC methods. To do this, several recursive procedures are run at the same time. They put the collected data to the same auxiliary table according to which at the end of simulation PR is computed.

Also it is possible to organize parallel PR computation with the MC method and the new information collection from the Web as the Crawler works independently of the computational algorithms.

The level computation is organized according to Algorithm 4. To implement Algorithm 4 two auxiliary tables are created to store the sets S_1 and S_2 . The levels of pages are stored in the table "pages". The root page can be either the page from which the crawling is started or it could be a page given by the user.

4.5 Graphical User Interface

The Graphical User Interface (GUI) is developed using Oracle Developer. All the functions of the interface are realized in the kernel of DBMS and are run on the server, which implies fast access to the data and the fast data analysis.

At the moment of the present article writing the GUI functions are as follows.

There is a possibility to set different parameters of the Crawler and to run the Crawler from GUI (see Figure 2). You can set such parameters as an initial URL, the file extensions which the downloaded pages must have. Also there is a possibility do not collect the pages with the given parts of URL. For example, if you make an experiment on a Web site that you know, you might prefer not to collect technical pages of that site.



Figure 2: Crawler parameters.

The user can compute a number of the WG characteristics. There is a possibility to calculate PR with PI and MC methods (see Figure 3). The user can set the number of iterations. Then, after the algorithms execution is finished, PR as well as the relative error are displayed. At each algorithms iteration there is a possibility to delete the data and to start the calculations from the beginning. The user can find the numbers of incoming and outgoing links, calculate the levels of pages. Furthermore, the user is able to determine a SCC for a given root page.

S	sarch		SC	C P	review	Robet functio	ns					
MC :	l itera	tion	PI 1 it	eration PI	n iterations	MC n iteral	ions	Restart MC	Restart	n=		
So age	rt by I in/ou	et at link	Sort s leve	by MC	Rank PI	discrepanc	v n	1C PageRank	MC visites	MC relativ error wrt	ve URL E PI	
_	1277	35	1	003965711357	7237831425620	0009094913	.003819	3204384825537354	9651 1679	0369141639	99)http://www-sop.inia.fr/	
31	607	t1	3	002306915670	3038594195375	.0012430856	.002283	8580823326289162	820-1004	0099949854	54)http://www-sop.inria.fr/coprin/index.html	_
3322	111	6	13	001449322142	3112412981104	0008306759	.001369	4049457811181350	615/602	0951410860	50 http://www-sop.inria.fr/acacia/personnel/Fabien.Gandon/menu.html	
305	252	251	5	0014031424464	1240627677075	.0008370676	001494	5166933192601573	678: 657	0651211479	Phttp://www-sop.inia.fr/mmosa/mobility_list/threads.html	
7345	261	247	11	/001263687317/	5582745513095	0013901686	001194	2404992277193038	327-525	0549493671	1 http://www-sop.inia.fr/axis/cbrtools/manual/api/tree.html	
1346	261	247	11	001263687317	\$582745513096	0013901686	.001230	6446439660878921	4000841	0261478240	0 http://www-sop.inia.fr/acis/cbrtools/manual/api/AlNames.html	
715	132	134	18	.001263069195	0653078803505	0023975571	.001203	3475354123114509	0954529	.0472829674	M http://www.sop.inia.fr/prisme/biblio/index.html	
0654	245	P	8	001229196830	/654057307334	0014840996	.001203	34/535412311450/	0/5/529	10210310119	Whitp://www-sopuma.hr/epidaure/personnel/malandan	
7344	261	15	11	001200116843	305914308677	0013744447	.001228	3698849199398553	708: 540	0235419085	35 http://www-sop.inia.hr/avis/cbrtools/manual/api/packages.html	
2940	150	119	9	001166821341	9670345227084	0013382210	.001176	0504268585350096	790/517	0079095960	0 http://www-sop.inia.fr/todeo/avega/phd/phd-html/hode2.html	
0900	130	129	13	001119329076	8684518242574	0013213633	.001044	1144021819488770	6511459	0671962127	27 http://www.sop.inia.hr/robotvis/personnel/orothwei/morse/docs/CO	Clitable:
8168	213	2	8	001028906667	3616975591527	.0002530637	.000982	6959079359518842	9668 432	10449124895	6 http://www-sop.inia.fr/croap/figue/web/index.html	
996	316	213	6	001002420038	2068415998905	0012981188	.000932	6512089206950753	741-1410	1.0696003936	36 http://www-sop.inria.hr/coprin/logiciels/ALIAS/ALIAS-C++/node315.1	kmi
994	316	198	6	001002117893	3546462076600	0012980349	.000965	3987993821754430	66119120	0466203570	<pre>P0 http://www-sop.inia.fr/coprin/logiciels/ALIAS/ALIAS-C++(hode313.1</pre>	kml
526	251	251	6	000996445740	9849020183225	0008052271	.001021	3668117204685093	731-1449	0250099626	26 http://www-sop.inria.hr/nimosa/nobility_list/mailist.html	
87	252	2	14	000959402543	7239207021136	.0008072288	.000948	5745222437313327	585 417	0112862130	0 http://www.sop.inia.fr/minosa/personnel/Davide.Sanglorgi/noca.ht	1
8515	442	19	9	000930793109	5111797062244	0009355295	.000895	2550641823264870	668/094	0071060390	0 http://www-sop.inia.fr/galaad/logiciels/synaps/html/main.html	
9172	440	282	10	000923424646	1958779906369	.0009369676	.001065	4881974126890609	1125/464	1430149730	30 http://www-sop.inia.fr/galaad/logiciels/synaps/htm/Files.html	
3864	439	85	11	000916936975	2408761524842	0009360145	.000978	1463898436558107	582-1130	1.0667542222	22 http://www-sop.inria.hr/galaad/logiciels/synaps/html/classes.html	
9264	140	81	10	.000916583159	3727518846453	0009359012	.001044	1144021819488770	65119159	1391376674	M http://www.sop.inia.fr/galaad/logiciels/synaps/html/hierarchy.html	
9191	440	80	10	000916464082	5867132153408	0009358828	.000932	6512089206950753	741-1410	0176625967	57 http://www-sop.inria.hr/galaad/logiciels/synaps/html/annotated.html	
2818	P25	125	B	000916438001	1931548704979	0000436073	.000969	0473536590636636	814:1426	10574063410	0 http://www-sop.inia.tr/croap/centaur/butorial/man/bableofcontents	_Z.html
8185	R15	p	10	000901278193	657351422129	0002390836	.000932	6512009206950753	741-1410	1.0048094691	N http://www-scp.innia.tr/croap/higue/web/credits.html	
8184	214	p	8	1000001078195	rb1783b769856	10002390326	.000944	0250041514352592	2010/415	10476615775	s fittp://www.sop.inia.tr/croap/tigue/web/help.html	
9278	140	25	10	000695727487	0178223635799	0009326951	.001000	0939803051361784	50219440	1174090276	% http://www-sop.inria.hr/galaad/logiciels/synaps/html/globals.html	
192065	440	¥3	10	publies3171346	\$288777524520	10009323037	.000914	4531365515107812	2011/102	10238272200	u netp ()/www.sop.inia.tr/galaachogiciels/synaps/html/functions.html	_
9266	440	19	10	0008865138613	0025020740210	0009312858	000984	9706669820999210	658(#33	1110606500	0 http://www-sop.innia.hr/galaad/logiciels/synaps/html/namespaces.htm	4
9279	1440	18	10	1000884408033	9425218940316	0009309644	.000889	430/670438823767	5929 (1991	10056792266	o http://www-sop.insa.tr/galaac@ogiciels/synaps/html/namespacemem	pers.htm
423	p74	28	8	1000870313878	075083293152	0014784120	.000884	8812689515863032	206:089	10167380884	94 http://www.sop.innia.tr/epidaure/personnel/malandain/	
0655	p29	p	18	1000864156129	1356233728433	0014649454	.000825	7375337517373472	214/[963	10444579334	94 http://www-sop.inria.fr/epidaure/personnel/malandain/cours/Topolo;	sy/sid001

Figure 3: PageRank computation methods.

The URL search is also realized in the interface (see Figure 4). Namely, it is possible to search hyper-links by their URL or by a part of URL. Then, the interface displays lists of incoming and outgoing links for a given page, the page level, the PR of the page calculated with different methods.

	SCC	Previe	The second second									
page no			URL				Leve	PI Pag	aRank	MC PageRar	nk MC visite	s
1	http://www.s	op.inia.fr/					1	.0039657113	772378314256	00381932043846	1255373549 1679	-
131	http://www.s	op.inria.fr/coprin/inde	x.html				3	0023069156	080355941953	00228385808233	12628916281004	
9322	http://www.s	op.inria.fr/acacia/pers	connel/Fabien. Gandon/men	a html			13	.0014493221	281124129611	00136940494578	111813506602	
802	http://www.s	op. innia. fry'mimosa/mo	bility_list/threads.html				5	.0014031424	642486276770	00149451669331	926015736657	
7346	http://www.s	op.innia.fr/axis/cbrtoo	is,inanuai(api(AlNames.htm	a -			11	.0012636873	765827455130	00123064464396	608789214 541	
7345	http://www.s	op.innia.fr/axis/cbrtoo	isimanuaij'apijtree.html				11	.0012636873	765827455130	00119424849922	771930383 525	
0715	http://www.s	op.inria.fr/prisme/bibli	io,lindex.html				0	.0012630691	506530788035	00120334753541	231145090 529	311
10654	http://www-s	op.inria <i>ĥr(</i> epidaure/p	ersonnel/inalandain				10	0012291988	076540573073	00120334753541	231145090529	
7344	http://www.s	op.innia.fr/axis/cbrtoo	is/manual/api/packages.htm	el.			11	.0012001168	1393069143086	00122836988491	993985537540	
2940	http://www.s	op. Innia.En/rodeci/ave;	ae/phd/iphd-html/node2.htm	á)			9	.0011668213	136703452270	00117605042685	853500967 517	-
Find by	y no	Find by URL	Find by URL part	1	in/out lin	ks level	PI Pa	qeRank	MC Pag	eRank	Refresh	
	Etter (hannes	son invia fri		-	277 35	6	00396571	135772378314	25 003819320	3948255373		
7266	http://www.	t no koja felastelektor	isiwanual/ani/AlManas htt	d								
7346 9914	http://www.s	t op.inia <i>fr(axis</i> (cbrtoo op.inia <i>fr(axis</i> (cbrtoo	ls/manual/api/AlMames.htm ls/manual/api/aid.cbr.took.	l core.reasoner.R	suse.Nml					_		
7346 9914 9918	http://www.s http://www.s http://www.s	t op. inia fr/axis/cbrtoo op. inia fr/axis/cbrtoo op. inia fr/axis/cbrtoo	isimanuai/api/AllHames.htm isimanuai/api/aid.cbr.took. isimanuai/api/aid.cbr.took.	i core.reasoner.R core.reasoner.R	isuse.html seasoner.htm	d						
7346 9914 9918 0536	http://www.s http://www.s http://www.s http://www.s	t op. Imia fr/axis/cbrtoo op. Imia fr/axis/cbrtoo op. Imia fr/axis/cbrtoo op. Imia fr/axis/cbrtoo	is, imanual (api (All Hames Atm is, imanual (api (aid.ctr.tools. is, imanual (api (aid.ctr.tools. is, imanual (api (aid.ctr.tools.	l core.reasoner.R core.reasoner.R core.nenory.sa	suse.html sasoner.htm nilarity.Aggr	ni egationFct.N	mi					
7346 9914 9918 0536 0570	http://www-s http://www-s http://www-s http://www-s http://www-s	t op. inia fr/axis/chrtoo op. inia fr/axis/chrtoo op. inia fr/axis/chrtoo op. inia fr/axis/chrtoo op. inia fr/axis/chrtoo op. inia fr/axis/chrtoo	işimanual api (All Kames htm İşimanual (api (aid.ctr.tools. İşimanual (api (aid.ctr.tools. İşimanual (api (aid.ctr.tools İşimanual (api (aid.ctr.tools	l core.reasoner.R core.reasoner.R core.reasoner.C	ause Mmi aasoner Mm nitarky Aggr aseSetRetri	ni egabionFct. N eveResult.ht	oral ml					
73%6 9914 9918 0536 0570 ncoming lin	http://www.s http://www.s http://www.s http://www.s http://www.s kts	t op. Innis Fr(axis)chroo op. Innis Fr(axis)chroo op. Innis Fr(axis)chroo op. Innis Fr(axis)chroo op. Innis Fr(axis)chroo	is, manuai (api (Ali Yames htm is, manuai (api (aid. cbr. tools. is, manuai (api (aid. cbr. tools. is, manuai (api (aid. cbr. tools. is, manuai (api (aid. cbr. tools.	il core.reasoner.R core.reasoner.R core.reasoner.C	teuse Jihni teasoner Jihr nilarky, Aggr JoseSetRebri	el egationFct.ht eveResult.ht	oni ni					
7346 9914 9918 0536 0570 ncoming lin 13150	http://www.s http://www.s http://www.s http://www-s http://www-s ks	t op. Innis Fr(axis/cbrtoo op. Innis Fr(axis/cbrtoo op. Innis Fr(axis/cbrtoo op. Innis Fr(axis/cbrtoo op. Innis Fr(axis/cbrtoo op. Innis Fr(axis/cbrtoo	is, Imanuai (api (Alf Kames Jihn Is) manuai (api (aid, chr. tools, Is) manuai (api (aid, chr. tools, Is) manuai (api (aid, chr. tools, Is) manuai (api (aid, chr. tools, reaction/DR: 15/CR/2001070	i core.reasoner.R core.reasoner.R core.reasoner.C 2.html	teuse Jihril teasoner Jihr nilarity. Aggr TaseSetRebri	ni egationFict. N evreResult. ht	oni ni					
7346 9914 9918 0536 0570 ncoming lin (3150 (3150	http://www.s http://www.s http://www.s http://www.s http://www.s http://www.s	t op. innia frifanisi(chrtoo op. innia frifanisi(chrtoo op. innia frifanisi(chrtoo op. innia frifanisi(chrtoo op. innia frifanisi(chrtoo op. innia frifanisi(chrtoo	is, inanuai (api/AliVanes.htm is (manuai/api/aid.cbr.tools. is, inanuai/api/aid.cbr.tools. is, inanuai/api/aid.cbr.tools. is, inanuai/api/aid.cbr.tools. neation/DR:15/CR/2001070 2002/comto.html	l core: reasoner R core: reasoner R core: reasoner C core: reasoner C 2.html	teuse Jihni teasoner Jihn nlarky, Aggr Jose Set Rebri	ni egationFct.ht eveResult.ht	2ml ml		, 			
7346 9914 9918 0536 0570 ncoming lin (3150 (3150 (3061)	http://www.s http://www.s http://www.s http://www.s http://www.s http://www.s http://www.s http://www.s	t op. inia fr/axis/cbrtoo op. inia fr/axis/cbrtoo op. inia fr/axis/cbrtoo op. inia fr/axis/cbrtoo op. inia fr/axis/cbrtoo op. inia fr/oxis/cbrtoo op. inia fr/oxis/cbrtoo op. inia fr/oxis/cbrtoo	Ishnanusi(ap)Allifanes Ath Ishnanusi(ap)Allifanes Ath Ishnanusi(ap)Aid chr. toola Ishnanusi(ap)Aid chr. toola Ishnanusi(ap)Aid chr. toola nisation/DR: 15/CR/2001070 2002/comte Ath Anno Ath Ann	i cons. reasoner .R cons. reasoner .R cons. reasoner .C cons. reasoner .C 2.html	teuse Jihni teasoner Jihn nlarky, Aggr JoseSetRebri	ni egetionFct.ht eveResult.ht	oral ral			-		
7346 9914 9918 0536 0570 ncoming lin (3150 N061 N061 N031 23848	http://www.s http://www.s http://www.s http://www.s http://www.s http://www.s http://www.s http://www.s http://www.s http://www.s	t op. imis fr(avis/chrtoo op. imis fr(avis/chrtoo op. imis fr(avis/chrtoo op. imis fr(avis/chrtoo op. imis fr(avis/chrtoo op. imis fr(opriv)noc op. imis fr(opriv)noc op. imis fr(avis/chrto	ishnaruai(ap)Allianes Atn ishnaruai(ap)Allianes Atnobi ishnaruai(ap)ai ad chr. toolo ishnaruai(ap)ai ad chr. toolo ishnaruai(ap)ai ad chr. toolo nisation(DR:E)(CR)2000.070 2002/comte Atnil conferentia. Mml chifformationg.com	il core, reasoner R core, reasoner R core, reasoner C core, reasoner C 2. html omplementaires.	ieuse.html ieasoner.htm nlarky.Aggr iaseGetRebri iaseGetRebri	el egationFict. N eveResult. ht	oni ed			•		
7346 9914 9918 0536 0570 ncoming lin (3150 94061 94051 23848 23836	http://www.s http://www.s http://www.s http://www.s http://www.s http://www.s http://www.s http://www.s http://www.s http://www.s	t op. hris fr(axis/cbrtoo op. hris fr(axis/cbrtoo op. hris fr(axis/cbrtoo op. hris fr(axis/cbrtoo op. hris fr(axis/cbrtoo op. hris fr(axis/cbrtoo op. hris fr(oprh/)hpc op. hris fr(oprh/)hpc op. hris fr(oprh/)hpc op. hris fr(oprh/)hpc op. hris fr(oprh/)hpc op. hris fr(oprio)hpc	Ishnaruai(api/AllKenes ktm Ishnaruai(api/AllKenes ktm Ishnaruai(api/aid.ctr.tools. Ishnaruai(api/aid.ctr.tools. Ishnaruai(api/aid.ctr.tools. resetten/DRI:tS/CR/2001070 2002/comte ktml resisten/DRI:tS/CR/2001070 2002/comte ktml resisten/DRI:tS/CR/2001070 2002/comte ktml resisten/DRI:tS/CR/2001070	il core, reasoner R core, reasoner R core, reasoner C 2. html omplementaires (consulado html	ieusie Jihril Deasoner Jihr Illarky, Aggr JaseSetRebri Intral	ni egationFrit. Ini erveResult. Ini	on) ed					
7346 9914 9918 0536 0570 ncoming lin (3150 94061 94051 (3848 (3836 8440)	http://www.s http://www.s http://www.s http://www.s http://www.s http://www.s http://www.s http://www.s http://www.s http://www.s	t op. hris /r/axis/chroa op. hris /r/axis/chroa op. hris /r/axis/chroa op. hris /r/axis/chroa op. hris /r/axis/chroa op. hris /r/seni/lorga op. hris /r/seni/lorga op. hris /r/galadjeni op. hris /r/pojc.gedrer op. hris /r/opic.gedrer	is, Imarua (ap) All Kanes Atm Binamual (ap) All Kanes Atm Binamual (ap) and ctr. tools. Binamual (ap) and ctr. tools. Is (Imarual (ap) and ctr. tools. Restlon/DR1: (5)/CR/2001070 2002) comts: html company. Signification (Amora Atm Company. Signification (Amora Atm) Company. Signification (Amora Atm) Company. Signification (Amora Atm) Company. Signification (Amora Atm) Company. Signification (Company. Signification (Company.	d cone, reasoner ,R core, reasoner ,R core, reasoner ,C 2.html complementaires , (consulado.html	isuse Jihni Isasoner Jihn Narky, Aggr IsaseSetRebri IsaseSetRebri	ni egationFct.hi eveResult.ht	on) ed					
73%6 9914 9918 0536 0570 ncoming lin 13150 90061 90061 90061 93848 23848 23848 23848	http://www.s http://www.s http://www.s http://www.s http://www.s http://www.s http://www.s http://www.s http://www.s http://www.s	t op. Imia <i>Fr(asis)</i> (cbrtos op. Imia <i>Fr(sorii)</i>)orgas op. Imia <i>Fr(sorii)</i> (orgas op. Imia <i>Fr(sorii)</i>)orgas op. Imia <i>Fr(sorii)</i> (orgas op. Imia <i>Fr(sorii)</i>)orgas op. Imia <i>Fr(sorii)</i> (orgas) op. Imia <i>Fr(sorii)</i> (orgas)	istmarusi(api/Allitenes htm istmarusi(api/Allitenes htm istmarusi(api/all.ctr.tools. istmarusi(api/all.ctr.tools. istmarusi(api/all.ctr.tools. resteter/DR:15/CR/2001070 resteter	l cont. reasoner R cont. reasoner R cont. reasoner C 2. html consplementaires. (consulado html	isuse Ahmi Isasoner Ahr Norky, Aggr IsasoSetRebri IsasoSetRebri	el egationFct. N eveResult. ht	oni mi					
7346 9914 9918 0550 ncoming lin 13150 9061 90631 23946 23856 wutgoing lin	bitp://www.s http://www.s http://www.s http://www.s http://www.s http://www.s http://www.s http://www.s http://www.s http://www.s http://www.s	t op. Inite // auto/christo op. Inite // forenti/organ op. Inite // forenti/organ/organ op. Inite // forenti/organ/o	Istinarual (api/Alkames http: Istinarual (api/alkames http: Istinarual) (api/alk.ctr.tools, Istinarual) (api/alk.ctr.tools, Isti	l cons. reasoner .R cons. reasoner .R cons. reasoner .C 2. html 2. html omplementaires. (cons.dado.html	isuse.html isasoner.htm nilarky.Aggr isasSetRebri html	el egationFct. N eveResult. In	oni ni					
7346 9914 9918 0550 0550 0550 0550 0550 0550 0550 05	http://www.s http://www.s	t op, inis /r/axis/chrico op, inis /r/axis/chrico op, inis /r/axis/chrico op, inis /r/axis/chrico op, inis /r/axis/chrico op, inis /r/conti/joc op, inis /r/conti/joc op, inis /r/conti/joc op, inis /r/conti/joc op, inis /r/contin/joc op, inis /r/contin/joc	ici, inorusi (apiAlifares hin ili, inorusi) apiala de totolo ili, inorusi (apiala de totolo ili, inorusi) apiala de totolo ili, inorusi (apiala de totolo ili, inorusi) (apiala de totolo estativa (1911) (2000) (2000) 2000) conte hini mi impiai hini deformation (rematione , arico Aruya) (datos, chienos anito, f., attrali detecho, f., attrali	il core, reasoner R core, reasoner R core, reasoner C 2. html complementaires, (consulado.html	inuse.html inasoner.htm nilarky.Aggr isseSetRetri html	n) egabionFct. N eveResult. ht	ord nl					
7346 9914 9918 0536 0536 0536 0536 0536 0536 0536 0536	Http://www.strain.com/ http://www.strain.com/	t pp. Intels Fr/astrafyChritoso opp. Intels Fr/astrafyChrito	kinnerusifacilatienes hin kinnerusifacilatienes hin kinnerusifacilatienes hin kinnerusifacilatienes kinnerusifacilatienes kinnerusifacilatienes reskontifiketti/Ketzoottoro zozozones hin anginacias hin hin hin hin hin hin hin hin	il core, reasoner & core, reasoner & core, reasoner .C 2. html corplementaires, (consulado html	ieuse Jihril ieasoner Jihr Inlarky, Aggr Jacobs GetRebri Jihrail	ni egabiorFct.hi eveResuit.ht	oni ni			• • •		

Figure 4: Search and Analysis Page.

5. PERFORMANCE TEST

We performed several experiments on www.inria.fr site with the following volume of data: 300.000 pages and 1.500.000 links. The iterations of the PI method is compared with the iterations of the MC method for the PageRank computation. Here we discuss the results of PageRank computation as the computation of this Web Graph characteristic is the most demanding. Therefore we carry out the performance test of our tool on the PageRank computation algorithms. The computation of one PI iteration took approximately 20 minutes and one iteration of the MC method took about 20% less time. For this test we have used a computer with Intel Pentium III 1 Ghz Processor and 512 Mbytes of RAM.

6. CONCLUSIONS

We presented "Web Graph Analyzer" software tool, which is design to perform a thorough study of the World Wide Web hyper-text structure. In particular, "Web Graph Analyzer" allows one to investigate local graph characteristics such as in-/out-degrees and page level as well as global characteristics such as PageRank and Giant Strongly Connected Component. The graph analysis is performed with the help of a user friendly GUI. All numerical algorithms are efficiently realized using stored procedures of Oracle DBMS. The system scales well with large volumes of data.

In the future versions of the "Web Graph Analyzer" we plan to include the algorithms for more graph characteristics and to introduce the possibility of displaying plots.

7. REFERENCES

- A.V. Aho, J.E. Hopcroft, and J.D. Ullman, *Data Structures and Algorithms*, Addison-Wesley, 1983.
- [2] K. Avrachenkov, N. Litvak, D. Nemirovsky, and N. Osipova, "Monte Carlo Methods in PageRank computation: When one iteration is sufficient", University of Twente Technical Report no. 1754, March 2005, also submitted for journal publication.
- [3] P. Berkhin, "A Survey on PageRank Computing", Internet Mathematics, v.2, no.1, pp.73-120, 2005.
- [4] L.A. Breyer, "Markovian page ranking distributions: some theory and simulations", 2002, available at http://www.lbreyer.com/preprints.html
- [5] S. Brin, L. Page, R. Motwami and T. Winograd, The PageRank citation ranking: bringing order to the Web, Stanford University Technical Report 1998, available at

http://dbpubs.stanford.edu:8090/pub/1999-66.

- [6] M. Henzinger, "Hyperlink Analysis on the World Wide Web", in Proceedings of The 6-th ACM Conference on Hypertext and Hypermedia (HT'05), September 2005, Salzburg, Austria.
- [7] A.N. Langville and C.D. Meyer, Deeper Inside PageRank, Internet Mathematics, v.1, no.3, pp.335-380, 2003.