EASYSEARCH UTILITY FOR LOW-BANDWIDTH, LOW-CONNECTIVITY COMMUNITIES

Ms.Reena R. Lokare¤
M.Tech (Comp Engg)
Mr.S.M.Sabale±
Asst.Professor,
Dr.B.A.T.U.Lonere,
District: Raigad,402103,India
¤reenalokare2002@yahoo.co.in
±sabale_sm@yahoo.com

ABSTRACT
The Internet has potential to deliver information to communities around the world that have no other information resources. But high telephone and ISP fees in combination with low bandwidth makes it unaffordable for many people to browse the web online. All existing search engines are designed for high bandwidth internet connections. Again, existing search engines overwhelms user by sending too many results than the user can handle. Information poor people like student or people in developing countries can't decide which links are useful to them. To overcome these limitations, we have proposed an EasySearch utility for low bandwidth, low connectivity regions. This utility makes use of existing search engines to get the links, filters result pages for getting exact content which is achieved by searching the files with their names and removing unwanted images. Our utility differs from existing search engines, in that it is designed to return low bandwidth results and it not only returns links to the pages but it reads the page and writes it to the server's memory so that whenever search is made for the second time, result pages are returned from the server's memory and server doesn't connects to the Internet, which saves bandwidth.

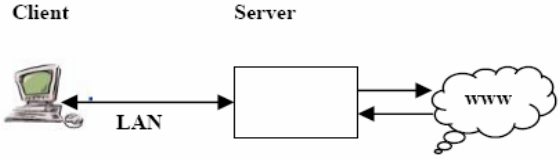
KEYWORDS

EasySearch utility, low connectivity communities, low-bandwidth, information processing.

I] Introduction

To make the information widely and easily available to the global community, we have proposed a new EasySearch utility here which is designed specially for low-bandwidth regions [1]. This utility has two-tier architecture where server has high speed Internet connection and clients are machines connected to the server through the high speed LAN. Client sends the query to be searched to the server over the LAN. Server stores the results of user's query in its memory, so it checks if the page is available in its memory and if present, returns to the client. If page is not available in server's memory, server connects to the Internet, invokes the search engine selected by the user, fetches the links, opens that links for reading contents and writes the content in its memory. Because of saving result pages in the server's memory, future requests are serviced from the server's local memory only and bandwidth gets saved. The main difference between EasySearch utility and existing search engines is that this search utility returns low bandwidth results whereas all existing search engines are developed for giving high bandwidth results. Our utility optimizes the result pages returned by existing search engine by removing duplicate contents and removing images. Another difference between EasySearch utility and existing search engines is that existing search engines returns only the links to the pages in the web servers, but EasySearch utility not only returns the links, but also the contents of the result pages[6],[7]. The main purpose behind developing this search utility is to produce low bandwidth results and give accurate contents to the user.

EasySearch utility:



EasySearch utility has following features:

- Low-connectivity, in that it does not rely on end to end connection every time.
- Low-bandwidth, in that it maximizes "information density" and only sends the exact and relevant contents.
- User-friendly, in that it does not overwhelm users with more results than they can manage.
- Similar to other search utility tools so that the skills that user acquires can be transferred to other web tools in the future.

The resulting system has three components:

1. **Client:**

Client provides graphical user interface for entering the query and viewing the results. The client provides local web pages that are viewed with a browser. All interfaces are viewed in the web browser as if client is reading from the Internet. While entering the query, client can also enter the number of pages and can select the search engine from where result pages should be retrieved.

2. Server:

The server searches for the requested web pages, either from its local memory or from the internet and returns the personalized, low-bandwidth, distinct results to the client. When server receives the query, it either searches the pages in its local memory or if not present in local memory, it invokes the existing search engines [2],[3].

There are two primary ways in which this server differs from other existing search engines:

- It keeps all the past results in his memory. This utility is specially designed for students who generally repeat the same search again and again. This is important in low-bandwidth and low-connectivity regions as bandwidth is limited, connectivity is less.
- It filters contents from the resulting pages, optimized for speed rather than response time.

Server of EasySearch utility does post processing on the result returned by the existing search engines. Following post processing techniques are used:

1. Removing duplicate pages: duplicate pages are totally removed from the resulting set and only unique pages are returned to the client.
2. Removing images: images are also completely removed from the result pages in order to save bandwidth.

3. File manager:

File manager is the program for managing stored files called as result pages. It provides the graphical interface where user can choose the files to be deleted.

II] Experimental Results:

In this chapter, results of EasySearch utility and Google, Clusty are shown in the graphical format[5]. Two main parameters are considered here for measuring results i.e. Time to search and Bandwidth utilization. Both parameters are again distinguished under two cases: Initial search and Repeated search.

We have distinguished the queries made to the EasySearch utility depending on the type of file searched. Generally html and pdf files are retrieved frequently from the Internet. But in addition to that, there are different file types like, Adobe file formats: pdf (portable document format) and ps(postscript), Lotus file formats: Lotus 1-2-3(wk1,wk2,wk3,wk4,wk5,whi,wks,wku), Lotus wordpro (lwp),Macwrite (mw),Microsoft file formats: EXCEL(xls), PowerPoint(ppt),word(doc),works(wks,wps,wdb),write(wri), Rich text format(rtf), Shockwave flash(swf) , Text(ans, txt), HTML.

Here, we are considering three types query that are distinguished depending on the type of file they are retrieving i.e. pdf & ps (Adobe), xls,ppt,doc(Microsoft), html.

Here we are showing results only for html file type queries.

Following are the tables of values of two parameters containing all the above cases:

1] Values for search times in milliseconds:

1.Values for EasySearch utility:

Search Engine	Type of query	No. of queries	Time(ms)	
			Initial query	Repeated query
Google	Html	10	143010	43
		20	2860200	80
		30	4290300	110
		40	5720400	203
		50	7150500	227
Clusty	Html	10	156798	46
		20	3135960	86
		30	4703940	121
		40	6271920	231
		50	7839900	234

2.Values for Google:

Search Engine	Type of query	No. of queries	Time(ms)	
			Initial query	Repeated query
Google	Html	10	2462	211.204
		20	4926	395.400
		30	7385	470.614
		40	9846	544.856
		50	12200	699.1034

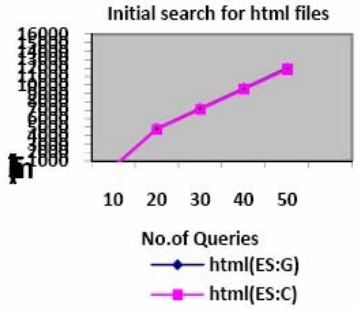
3.Values for Clusty:

Search Engine	Type of query	No. of queries	Time(ms)	
			Initial query	Repeated query
Clusty	Html	10	2567	234.234
		20	5078	456.800
		30	8756	654.456
		40	10624	738.628
		50	13280	822.285

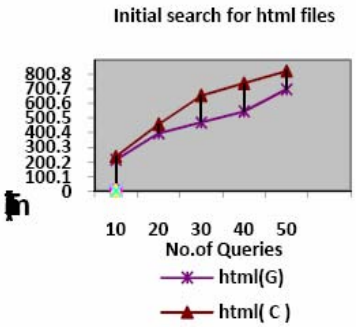
Following graph plots search times of various search engines . Here, for each type of query, graphs have been plotted for showing search time for initial query and repeated query.

A] Initial search:

1.Initial search for html files using EasySearch utility:

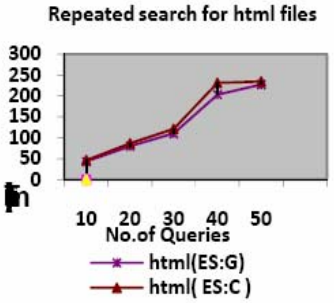


2.Initial search for html files using Google & Clusty search engines:

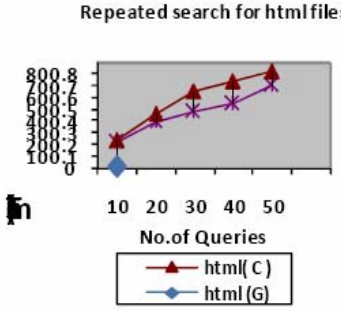


B] Repeated search:

1. Repeated search for html files using EasySearch utility:



2.Repeated search for html files using Google & Clusty search engines:



2] Values for Bandwidth utilization in %:

1.Values for EasySearch utility:

Search Engine	Type of query	No. of queries	Bandwidth utilization(%)	
			Initial query	Repeated query
Google	html	10	239	3.0
		20	4780	38
		30	7170	45
		40	9560	65
		50	11950	72
Clusty	html	10	237	4.2
		20	4740	45
		30	7110	49
		40	9480	69
		50	11850	79

2.Values for Google:

Search Engine	Type of query	No. of queries	Bandwidth utilization(%)	
			Initial query	Repeated query
Google	html	10	110	8.6
		20	2132	80
		30	3312	108
		40	4165	167
		50	5311	180

3.Values for Clusty:

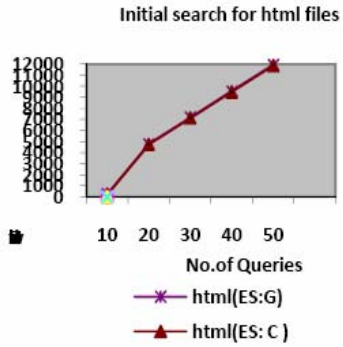
Search Engine	Type of query	No. of queries	Bandwidth utilization(%)	
			Initial query	Repeated query
Clusty	html	10	122	9.1
		20	2344	86
		30	3254	114
		40	4256	172
		50	5412	194

Following graph plots bandwidth utilization of various search engines.

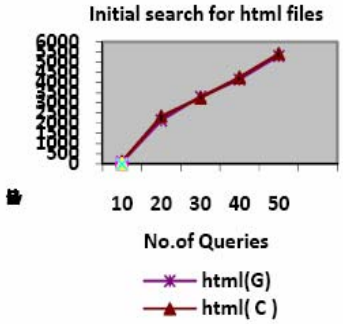
Here, for each type of query, graphs have been plotted for showing bandwidth utilization for initial query and repeated query.

A] Initial search:

1. Bandwidth utilization for html files using EasySearch utility for initial search:

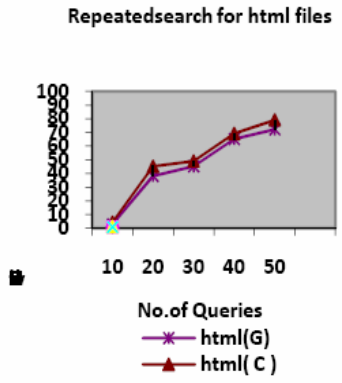


2. Bandwidth utilization for html files using Google, Clusty search engines for initial search:

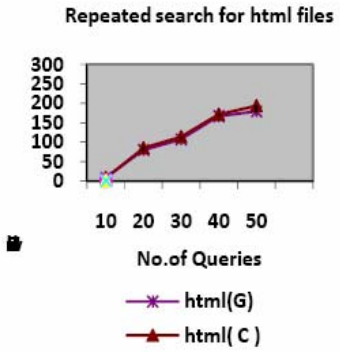


B] Repeated search:

1. Repeated search for html files using EasySearch utility:



2.Repeated search for html files using Google & Clusty search engines:



Here results clearly shows that this search utility is better than other search engines when user searches query for the second time (i.e. repeated search).

III] Conclusion:

We have proposed here a search utility for low-bandwidth and low-connectivity regions which has different features compared to other search engines like, results are optimized for low bandwidth by removing duplicate pages and images[1]. This utility has two-tier architecture in which, server saves result pages of previous searches which saves bandwidth as in the case of repeated search, internet is not used which saves bandwidth.

References:

[1] William Thies, Libby Levison, "Searching the world wide web in low connectivity communities", Lab for computer science,M.I.T. Cambridge, MA 02138 USA.

[2] Elliote Rusty Harold, "Java Network Programming",O'Reilly,2006.

[3] Herbert Shield, "Complete Reference", Tata Mc.Graws Hill,2000.

[4] William Frakes and Richardo Baeza Yates. Information retieval. Data structures and algorithms, Prentice hall,1992.

[5] S. Brin and l.Page. The anatomy of a large-scale hypertextual web search engines in seventh International World Wide Web Conference, Brisbane, Australlia,1998.

[6] Libby Levison, Bill Thies, Saman Amarasinghe," The TEK Search Engine", MIT Laboratory for Computer Science.

[7] Libby Levison, William Thies, Saman Amarasinghe," Providing Web Search Capability for Low-Connectivity Communities", MIT Laboratory for Computer Science.