SMART CACHE FOR CENTRAL OFFLINE DOWNLOAD SYSTEM

Sidath Bandara and Gihan Dias
Department of Computer Science and Engineering.
Email: sbandara@cse.mrt.ac.lk, gihan@cse.mrt.ac.lk

SMART CACHE FOR CENTRAL OFFLINE DOWNLOAD SYSTEM

Sidath Bandara and Gihan Dias
Department of Computer Science and Engineering.
Email : sbandara@cse.mrt.ac.lk, gihan@cse.mrt.ac.lk

ABSTRACT

In Sri Lanka the communication links to the Internet are heavily congested during the peak hours. But they are not fully utilized during the off-peak hours such as in the night. Downloading a large file is an impossible task during peak hours. As a solution an offline downloading system was developed for FTP and HTTP which is available centrally to all Internet users in an Institution. It can be used more easily than existing products for the average Internet user. This paper discusses the need for an offline downloading system and about the offline downloading system that was implemented.

THE NATURE OF INTERNET ACCESS AND TRAFFIC

The major part of Internet traffic is used for Web access and file transfer through ftp. Most users of the Internet use it for Web browsing and downloading of files.

Normally browsing is the process of following hyperlinks and viewing different web pages. This process is highly interactive. A user clicks at a hyperlink in a web page and waits for the next web page to appear. These pages and other embedded objects such as images are normally not large in size. In contrast file downloading through the web is not as interactive as browsing. Normally a user selects files to be downloaded by browsing or searching. These files could be software applications, documents etc. Though most of the files are in compressed form their size can range from a few tens of kilo bytes to several hundred mega bytes.

In the process of file downloading through FTP (File Transfer Protocol), a user gets connected to a FTP server and traverses the directory structure searching files. Again this process of traversing and viewing files in the directories is a highly interactive process. User can start downloading a file once he selects a file to be downloaded. These files are normally large in size.

PROBLEMS WITH CONGESTED INTERNET COMMUNICATION LINKS

Communication links to the Internet could be heavily congested specially during peak hours. As a result while accessing Internet through FTP or Web, the connection to the remote server may get lost due to timeouts or connection reset by remote server. Large files take a considerable amount of time to download. During this time there is a high probability of losing the connection to the remote server. In these conditions downloading a large file is very difficult.

Consider the example of an organization that has a 64KBPS leased line to the Internet. Assume that at any given time during the peak hours of usage, there are 40 effective users of Internet. Assume all these users use only one instance of web browser (i.e. only one window for web access). Then a single user will have a transfer rate of just above 200 bytes/s. The real situation may be worse than this, particularly

when a user opens more than one window for Web browsing. Even at this rate, downloading a 10MB software application (this size is quite usual these days) will not be possible during the 8 hours on a working day. It will not be possible to download a file smaller than 10MB due to timeouts.

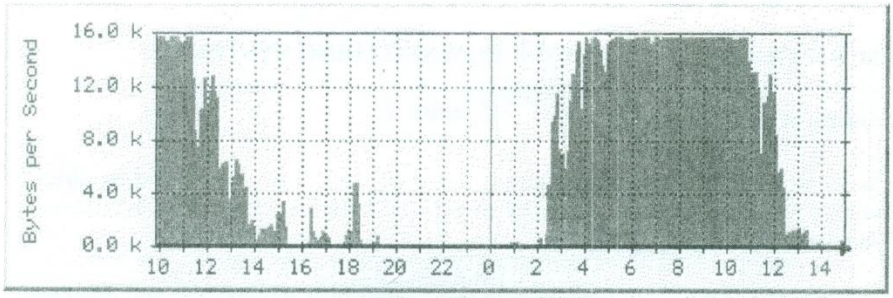
Apart from this, downloading large files during the peak hours will make the communication links even more congested. That is downloading large files during peak hours is not just a problem on its own but it will also make the Internet communication link more congested.

So a solution is necessary for easy downloading of large files and to make communication links less congested during the peak hours.

OFFLINE DOWNLOADING

The obvious solution to the above problem is to increase the bandwidth of the communication link to the Internet. But high bandwidth to the Internet costs millions of Rupees. Since Sri Lanka is a developing country with a lot of other social and political problems investing in very high bandwidth links is not possible and might not be justified. This is particularly relevant to academic institutions that depend on funds from government and other donor institutions.

But when we examine the traffic pattern of the Internet access during a day we can see that there is another feasible solution. The following diagram shows traffic to the Internet, collected from a router in the Lanka Educational and Research Network (LEARN). The bandwidth of this link is 128KBPS.

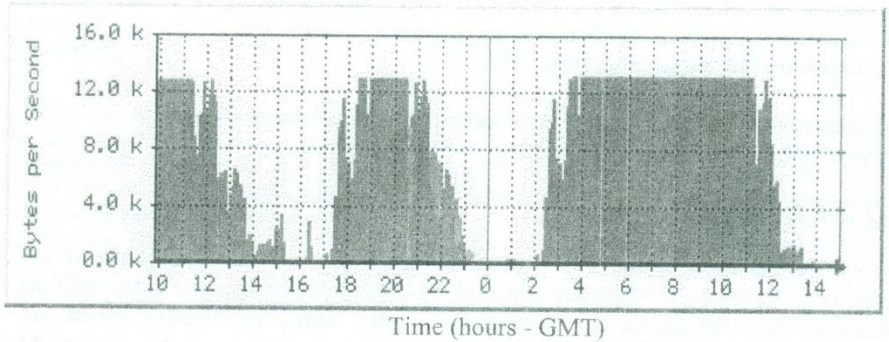


As seen in this diagram during the peak hours the link is fully utilized. But there is a particular time slot where the bandwidth is hardly utilized. This can be viewed as a waste, because expensive communication links are not fully utilized. During the peak hours everybody competes for the bandwidth. If there is a way to reduce the Internet traffic during the peak hours and/or re-schedule the Internet traffic during off-peak hours better results can be obtained.

As explained earlier Internet access could be divided into two major categories, interactive and non-interactive access. The non-interactive category is the category for

re-scheduling during the off-peak hours. In other words it is possible to reduce or stop large file downloads during peak hours and re-schedule them in off-peak hours.

The following diagram shows a possible traffic pattern for the above mentioned 128KBPS link, with the proposed Offline Download system operating properly.



Time (hours - GMT)

This method has multiple advantages.

1) As large files are downloaded during the off-peak hours there is a very good possibility to download that file without any problems. As the end result, users can download very large files without any difficulty, which is not possible during peak hours without the offline download system.
2) Better usage of expensive communication link as the non-utilized time slot is also put into use.
3) Tendency to make communication links less congested during peak hours, thereby increasing interactive performance of Internet link.

DESCRIPTION OF THE OFFLINE DOWNLOADING SYSTEM

Rationale of the Design

To schedule large file downloads during off-peak hours the user should be able to record his request. This can be done by writing the URL associated with the file to be offline downloaded in a log file.

Then there should be a software module that should read this log file and download each file represented by URLs during the off-peak hours. This module will store downloaded files in the secondary storage device locally so that user can fetch those files later.

Users use web browsers to browse the web. The easiest user interface for accessing FTP sites is also the web browser. Also there should be a method to schedule or request a particular URL to be offline downloaded. If this method is a web based one it will make the offline download system more usable. That is a user can schedule the file to be offline downloaded with very little effort while browsing. A web based method to request files to be offline downloaded can be built easily using a web server. In the user interface of this method there may be user inputs such as email address of the user and URLs. These should be stored in a log file as mentioned

earlier. To do this the web server should be capable of producing a log file containing details such as URLs and email addresses. That is the web server should be capable of server side processing.

When a user accesses a FTP site using a web browser he gets a very simple output in his web browser. That is he can view the files and directories in a particular directory of the FTP site. Due to this simple nature of the output it is very easy to modify this output to add another hyperlink which can be used to schedule a particular file to be offline downloaded. Normal proxy servers used for web and FTP caching can be modified to generate this kind of output when a user accesses a FTP site.

Based on these aspects the offline download system that was implemented consists of following modules.

1) A web server.
2) The URL Fetch module – A software module, which is capable of downloading files and sending emails.
3) A Modified version of Squid Object cache.

Fig: 1 shows the simplified architecture of the system that was developed for Offline Downloading files.

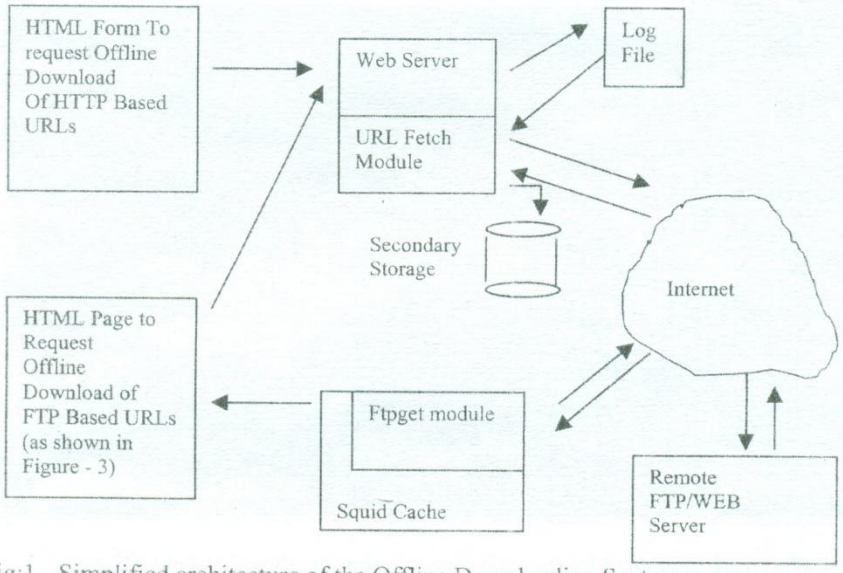


Fig:1 - Simplified architecture of the Offline Downloading System

The following sections describe the purpose of each of the module.

Web Server

The web server is capable of server side processing and produces dynamic web pages. It is somewhat similar to a web server that can handle CGI scripts. But this particular web server is a Java based web server that supports the Servlet API [1].

There are several functions of this web server

a) When a user requests for a file to be downloaded he will be given a Web form to be filled. The information gathered (the URL of the file, email address to inform the result of the attempted download) should be logged in a file. So creating dynamic web pages for user interaction and logging of URLs is one major function of this web server. For this reason a web server capable of server side processing is used.

b) As mentioned above URLs are kept in a log file. During the off-peak hours these URLs will be downloaded to the hard disk of the server. When a user wants to fetch this file he can use the web server to fetch it. That is this web server is used to create a web page consisting of a list of all the files which were downloaded offline. This is shown in the Fig: 2.

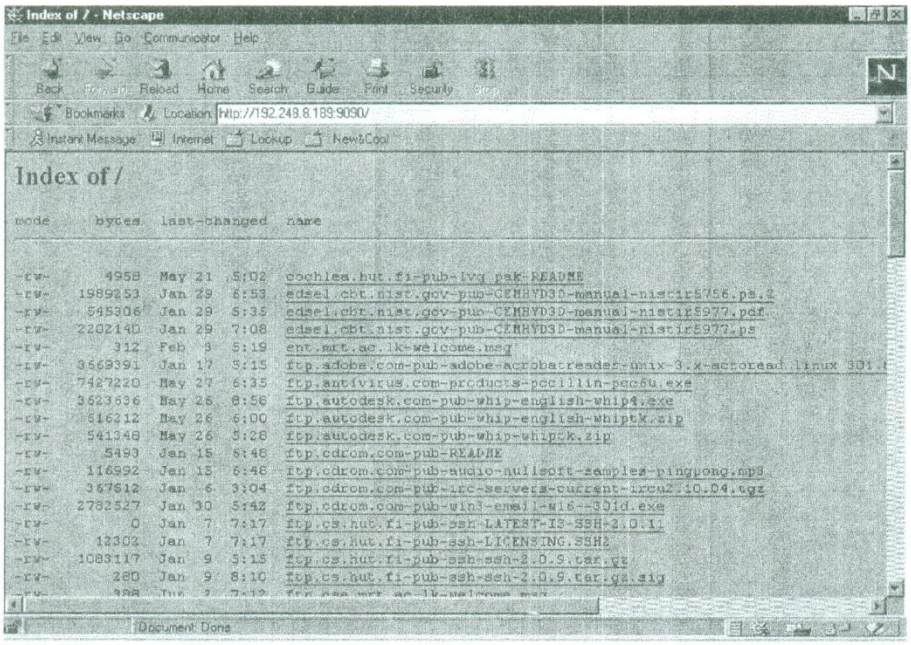


Fig: 2 – Web page generated by the web server listing the downloaded URLs

c) Apart from the above tasks, web server also checks the local storage when a user requests for a URL to be downloaded. If this file represented by the URL is already in the hard disk he will be given the option to fetch that file which is in the local storage.

URL Fetch Module

As described above, the requested URLs for offline downloading and an optional email address is stored in a file by the web server. The URL fetch module reads this file and tries to download these URLs during the off-peak hours. If the file download is successful an email will be sent to the user if the email address is available. If the downloading is not possible due to an error (eg. Incorrect URL, remote server can't be contacted) an email will be sent informing of the problem. The files downloaded by this module will be saved in a location where the user can access the files using the web server. If a file could not be fetched due to time-outs it will be retried several times by this module. If the file to be downloaded is already in the hard disk that file will not be downloaded again. Instead an email will be sent to user.

Modified Version of Squid Cache

Squid [2] is a popular object cache used as an Internet proxy server. To make FTP offline downloading easy this Squid cache was slightly modified. This modification is limited to the 'ftpget' module of the Squid cache. The original ftpget module communicates with a remote ftp server and produces a directory listing of each directory as a HTML page. The Ftpget module was modified to produce a HTML page with an additional hyperlink called 'Offline Download' for each file in the directory. Once the user clicks this 'Offline Download' hyperlink the URL associated with the particular file will be sent to the above mentioned web server. This is done by URL encoding. At this point the web server has the ftp URL to be downloaded. An example output of the modified ftpget is shown in Fig: 3.

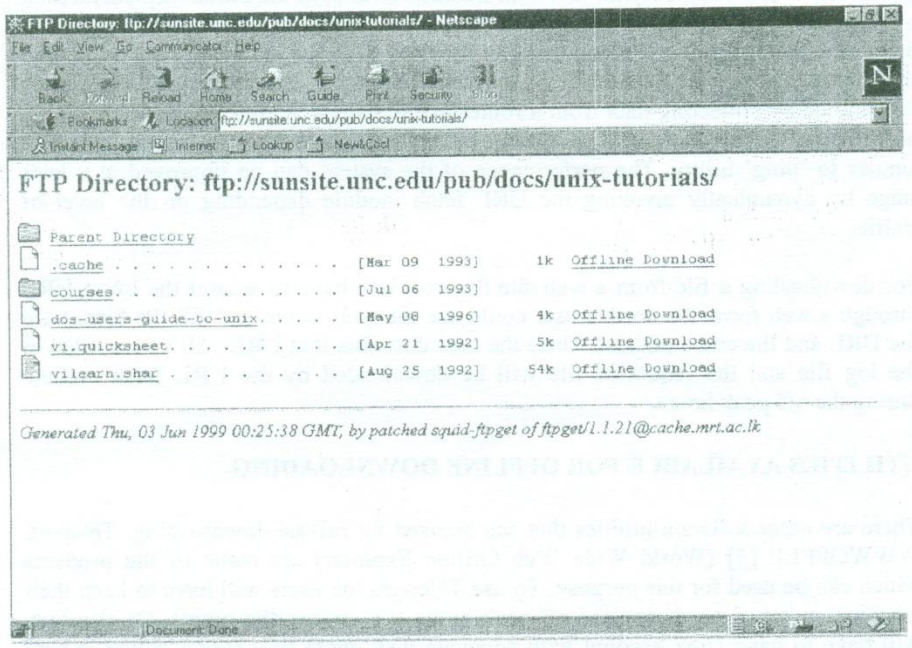


Fig: 3 – User interface when accessing ftp sites

Pattern of Events

Consider the example of the way a user requests a file to be downloaded through the Offline Download system. First the user will type the FTP site name in his web browser at the relevant text box. The user should have set the proxy configuration in his browser. This is done by setting the appropriate IP and port of the Squid cache server. Then once the user access the FTP site the Squid cache (ftpget module) will generate HTML pages for the user. An additional hyperlink (shown as 'Offline Download') exists for each file in these pages. The URL of the file to be offline downloaded is url-encoded in this hyperlink itself. Once a user clicks at this hyperlink a request will be sent to the web server. Then the web server will give the user a web form and ask for an email address from the user. This address will be used to inform the result of the offline download to the user. Once the user confirms that he wants to offline download the file, the request will be written to a log file.

During the off-peak hours (i.e. normally in the night) the URL fetch module will start its work. The above mentioned log file will be read by this URL fetch module. This module downloads files and sends emails to relevant users as described in the section 'URL Fetch Module'. This describes the events when the user request a file to be offline downloaded from a FTP site.

According to statistics the traffic in the Internet communication link of Moratuwa University is low from 11 p.m. to 4 a.m. Because of this, in the initial implementation of this system the URL fetch module is invoked at 11 p.m. A better approach is to determine the traffic level of Internet communication link dynamically and to invoke the URL fetch module depending on the level of traffic. The level of traffic can be determined by collecting data from a router using SNMP. A simpler approach would be to calculate the round trip time of a packet to remote Internet nodes, which is similar to 'ping' utility. The performance of the system can be improved at a later stage by dynamically invoking the URL fetch module depending on the level of traffic.

For downloading a file from a web site the user will have to submit the exact URL through a web form. To do this user could use the web server and fill the form with the URL and the email address. Once the user does this that URL will be recorded in the log file and the requested file will be downloaded by the URL fetch module during the off-peak hours.

UTILITIES AVAILABLE FOR OFFLINE DOWNLOADING

There are other software utilities that can be used for offline downloading. Teleport, WWWOFFLE [3] (World Wide Web Offline Explorer) are some of the products which can be used for this purpose. To use Teleport, the users will have to keep their machine switched on during the off-peak hours (i.e. during the night). Or else they will have to have Unix account with adequate disk quota in a server which is kept switched on in the night. Most of the users in our academic institutions will not be able to get these facilities. Even if they have these facilities this is not as easy as using the proposed Offline Download system when accessing Internet.

When using WWWOFFLE user will have to switch this utility to online and offline mode manually. Hence it is difficult to use WWWOFFLE for offline downloading a particular set of URLs in the multi user environment.

CONCLUSION

The aim of this project was to implement a system for Offline downloading with minimum of hassle. For FTP we have achieved this, as user can schedule files to be offline downloaded in a central server simply by clicking the mouse. To offline download files from a web server user will have to cut and paste or type the exact URL in a HTML form. Hence scheduling files for offline download from a web site is not as easy as for FTP. But with the experience of this project a better system may be implemented. The System also permits frequently access files to be downloaded only once.

Reference:

1 Servelet Development Kit API Documentation -
http://java.sun.com/products/servlet/index.html

2 Squid Object Cache Documentation -
http://squid.nlanr.net/Squid/documentation.html

3 WWWOFFLE - World Wide Web Offline Explorer Documentation -
ftp://sunsite.unc.edu/pub/Linux/apps/www/servers/wwwoffle-2.4.tgz