I.J. Information Technology and Computer Science, **2013, 11, 42-53**
Published Online October 2013 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijitcs.2013.11.05

A Proposed Model for Web Proxy Caching Techniques to Improve Computer Networks Performance

Nashaat el-Khameesy
Prof. and Head of Computers & Information systems Dept- SAMS, Maady Cairo, Egypt
E-mail: Wessasalsol@gmail.com

Hossam Abdel Rahman Mohamed
Computer & Information System Dept- SAMS, Maady Cairo, Egypt
E-mail: Hrahman@Transit.com.eg, Habdel@ENR.gov.eg

***Abstract*—** one of the most important techniques for improving the performance of web based applications is web proxy caching Technique. It is used for several purposes such as reduce network traffic, server load, and user perceived retrieval delays by replicating popular content on proxy caches that are strategically placed within the network. The use of web proxy caching is important and useful in the government organizations that provides services to citizens and has many branches all over the country where it is beneficial to improve the efficiency of the computer networks performance, especially remote areas which suffer from the problem of poor network communication. Using of proxy caches provides all users in the government computer networks by reducing the amount of redundant traffic that circulates through the network. It also provides them by getting quicker access to documents that are cached. In addition, there are a lot of benefits we can obtain from the using of proxy caches but we will address them later. In this research, we will use web proxy caching to provide all of the above benefits that we are mentioned above and to overcome on the problem of poor network communication in ENR (Egyptian National Railways). We are going to use a scheme to achieve the integration of forward web proxy caching and reverse proxy caching.

***Index Terms*—** Web Proxy Caching Technique, Forward proxy, Reverse Proxy

I. Introduction

One of the most well-known strategies for improving the performance of Web-based system is Web proxy caching by keeping Web objects that are likely to be used again in the future in location closer to user. The mechanisms The Web proxy caching are implemented at the following levels: client level, proxy level and original server level. [1], [2] it is known that proxy servers are located between users and web sites for lessening of the response time of user requests and saving of network bandwidth. We also should build an efficient caching approach in order to achieve better response time.

Generally, we use web Proxy servers to provide internet access to users within a firewall. For security reasons, companies run a special type of HTTP servers called "proxy" on their firewall machines. When a Proxy server receives any requests from the clients, it forwards them to the remote servers intercepts the responses, and sends the replies back to the clients. Due to we use the same proxy servers for all clients inside of the firewall in the same organization, these clients share common interests and they would probably access the same set of documents and each client tends to browse back and forth within a short period of time, So this provides the effectiveness of using these proxies to cache documents. Therefore, this will increase the hit ratio for a previously requested and cached document on the proxy server in the future. In addition to web caching at proxy server saves network bandwidth, it also provides lower access latency for the clients.

Most Web proxy servers are still based on traditional caching policies. These traditional caching policies only consider one factor in caching decisions and ignore the other factors that have impact on the efficiency of the Web proxy caching. Due to this reason these conventional policies are suitable in traditional caching like CPU caches and virtual memory systems, but they are not efficient in Web caching area. [3], [4]

We use the proxy cache of the proxy server and it is located between the client machines and origin server. The work of the proxy cache is similar to the work of the browser cache in storing previously used web objects. The difference between them is the browser cache which deals with only a single user, the proxy server services hundreds or thousands of users. The work of the proxy cache is as follow, when the proxy server receives a request it checks its cache at first if the request is found the proxy server sends the request directly to the client but if the request is not found the proxy server forwards the request to the origin server and after the origin server replies to the proxy server it forwards the request to the client and also save a copy from the request in local cache for future use. The proxy caching is used to reduce user delays and to reduce Internet congestions it is widely utilized by computer network administrators, technology providers, and businesses. [5], [6], [7]

The proxy server uses its filtering rules to evaluate the request, so it may use IP address or protocol to filter the traffic. If the request is valid by the filter, the proxy provides the content by connecting to the origin server and requesting the service on behalf of the client in case the required content is not cached on the proxy server. The proxy server will return the content directly to the client if it was cached before by the proxy server

We must consider the following problems before applying web proxy caching:

Size of Cache: In traditional architectures each proxy server keeps records for data of all other proxy servers. This will lead in increasing in cache size and if cache size becomes large this will be a problem because as cache size is larger, Meta data become difficult to be managed. [8]

Cache Consistency: We should ensure that Cache Consistency is verified to avoid Cache Coherence problem. Cache Consistency means when a client send requests for data to proxy server that data should be up-to-date. [9]

Load balancing: There is must be a limit for number of connections to certain proxy server to avoid the problem of overloaded only one server than the other in case we use load balancing. [10]

Extra Overhead: When all the proxy servers keep the records of all the other proxy servers, this will lead to extra overload in the system which already produces congestion on all proxy servers. This extra head due to each proxy server in the system must check the validity of its data with respect to all other proxy servers. [11]

In addition to the proxy cache provide some advantages such as a reduction in latency, network traffic and server load, it also provides some more advantages

- Web proxy caching decreases network traffic and network congestion and this automatically reduces the consumption of bandwidth
- Web proxy caching reduces the latency because of the followings:

  A. When a client sends to the proxy server a request already cached in the proxy server so in this case the proxy server will reply directly to the client instead of send the request to the origin server.

  B. The reduction in network traffic will make retrieving not cached contents faster because of less congestion along the path and less workload at the server.

- Web proxy caching reduces the workload of the origin Web server by caching data locally on the proxy servers over the wide area network.
- The robustness and reliability of the Web service is enhanced because in case the origin server in unavailable due to any failure in the server itself or any failure in the network connection between the proxy server and the origin server, the proxy server will retrieve the required data from its local cache.
- Web caching has a side effect that allows us a chance to analyze an organization's usage patterns.

In addition to proxy caches provide a significant reduction in latency, network traffic and server load, they also produce set of issues that must be considered.

- A major disadvantage is the resend of old documents to the client due to the lack of proper proxy updating. This issue is the focus of this research.
- A single proxy is a single point of failure.
- A single proxy may become a bottleneck. A limit has to be set for the number of clients a proxy can serve.

Therefore, in all government institution those provide services to citizen, we must be searched about methods and solutions to enhance the efficient of services delivery , and as we know that most places away from Cairo state is facing failure in the network because the lack of infrastructure and possibilities of the services provider (ISP).

There has been a lot of research and enhancement in computer technology and the Internet has emerged for the sharing and exchange of information. There has been a growing demand for internet-based applications that generates high network traffic and puts much demand on the limited network infrastructure. We can use addition of new resources to the network infrastructure and distribution of the traffic across more resources as a possible solution to the problems of growing network traffic.

Using of proxy caches in the government computer networks is useful to the server administrator, network administrator, and end user because it reduces the amount of redundant traffic that circulates through the network. And also end users get quicker access to documents that are locally cached in the caches. However, there are additional issues that need to be considered by using of proxies. In this study, we will focus on Web proxy caching since it is the most common caching method.

1.1 Problem Statement

The governmental organizations which provide services to citizen must target efficient and more reliable services while keeping cost-effective criteria. Throughout this paper we consider the case of the Egyptian National Railway (ENR) datacenter which serve many applications supported to many branches spreaded allover Egypt which are quite faraway from Cairo state. Current infrastructure faces so many challenging problems leading to poor reliability as well as ineffective services and even more discontinuity of such services even at the headquarter datacenter. The main attributes of the problems facing the ENR network are summarized as following:

- A major problem of the remote site is unstructured and their heavy network traffic.
- The network overloading might result in the loss of data packets
- The origin servers loaded most of the time
- Transmission delay – normal traffic data but low speed of the line.
- Queuing delay due to huge network traffic
- Slow the services that provided to citizens.
- Processing delay – due to any defection of the network device
- Network faults can cause loss of data
- Broadcast delay – due to presence of broadcasting on network

II. Proxy Caching Overview

Caches are often deployed as forward proxy caches, reverse proxy caches or as transparent proxies.

2.1 Forward Proxy Caching

The most common form of a proxy server is a forward proxy; it is used to forward requests from an intranet network to the internet. [12]

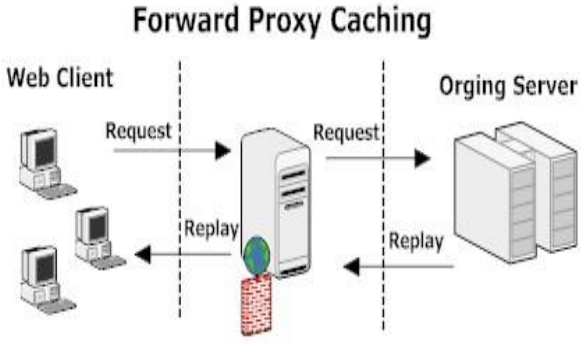


Fig. 2.1: Forward proxy caching

When the forward proxy receives a request from the client, the request can be rejected by the forward server or allowed to pass to the internet or [13] retrieved from the cache to the client. The last one reduces the network traffic and improves the performance.

On the other hand, the forward proxy treats the requests by two different ways according to the requests are blocked or allowed. In case the request is blocked, the forward proxy returns an error to the client. In case the request is allowed, the forward proxy checks either the request is cached or not; if it is cached, the forward proxy returns cached content to the client. If it is not cached, the forward proxy forwards the request to the internet then returns the retrieved content from the intent to the client.

The above figure explains the work of the forward proxy in case the request is allowed but not cached on the forward proxy A. the forward proxy will send the request to the server on the internet then the server on the internet return the required content to the forward proxy and finally the forward proxy return the received content to the client and cached it on its cache for future and same request. The cached content on the forward proxy will reduce the network traffic in the future and actually improves the performance of the whole system.

2.2 Reverse Proxy Caching

The other common form of a proxy server is a reverse proxy; it performs the reverse function of the forward proxy, it is used to forward requests from an internet network to the intranet network. [14] This provides more security by preventing any hacking or an illegal access from the clients on the internet to important data stored on the content servers on the intranet network. By the same way, if the required content is cached on the reverse proxy, this will reduce the network traffic and improves the performance.[15]

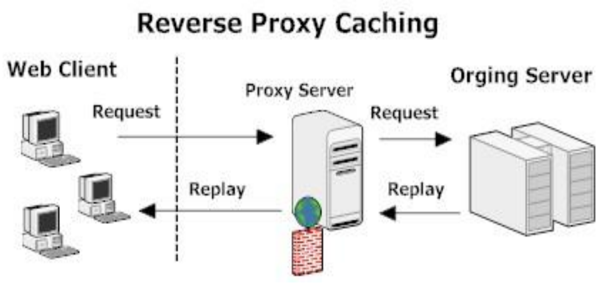


Fig. 2.2: Reverse proxy caching

The advantages of reverse proxy are

- Solving single point of failure problem by using load balancing for content servers.
- Reducing the traffic on the content servers in case the request is blocked by the reverse proxy. In this case the request is rejected directly by the reverse proxy without interrupt the content servers.

- Reducing the bandwidth consumes by blocked requests as it is blocked directly by reverse proxy before reaching to the content servers.

The function of the reverse proxy is the same as the function of the forward proxy except the request is initiated from the client on the internet to the content servers in the internal network. At first, the client on the internet sends a request to the reverse proxy. If the request is blocked, the reverse proxy returns an error to the client. If the request is allowed, the reverse proxy checks if the request is cached or not. In case the request is cached, the reverse proxy returns the content information directly to the client on the internet. In case the request is not cached, the reverse proxy sends the request to the content server in the internal network then resends the retrieved content from the content server to the client and also cached the content information from the content server for future requests to same content information [16]

2.3 Transparent Caching

Transparent proxy caching eliminates one of the big drawbacks of the proxy server approach: the requirement to configure Web browsers. Transparent caches work by intercepting HTTP requests and redirecting them to Web cache servers or cache clusters. [17] This style of caching establishes a point at which different kinds of administrative control are possible; for example, deciding how to load balance requests across multiple caches. There are two ways to deploy transparent proxy caching: at the switch level and at the router level. [18]

Router-based transparent proxy caching uses policy-based routing to direct requests to the appropriate cache(s). For example, requests from certain clients can be associated with a particular cache. [19]

In switch-based transparent proxy caching, the switch acts as a dedicated load balancer. This approach is attractive because it reduces the overhead normally incurred by policy-based routing. Although it adds extra cost to the deployment, switches are generally less expensive than routers. [20]

III. Proxy Caching Architecture

The following architectures are popular: hierarchical, distributed and hybrid.

3.1 Hierarchical Caching Architecture

Caching hierarchy consists of multiple levels of caches. In our system we can assume that caching hierarchy consists of four levels of caches. These levels are bottom, institutional, regional, and national levels [21]

The main object of using caching hierarchy is to reduce the network traffic and minimize the times that a proxy server needs to contact to the content server in the internet or in the internal network to provide the client with needed content information .These multiple caches works in that manner in case of forward proxy, at first the client initiate a request to the bottom level cache. If the needed content information is found and cached on it, it returns this information to the client directly. If this information is not cached on it, it will forward the client request to the next level cache that is institutional. If this cache found the needed information cached on it, it will return it to bottom level cache then the bottom level cache returns them to the client. If the needed information is not cached on it, it will forward the request to regional level. If the needed information is cached on it, it will return the needed information to the institutional level cache then the institutional level cache returns them to the bottom level cache and finally bottom level cache returns them to the client. If the needed information is not found not found on it, it will forward the request to the last level of cache that is national, if the needed information is found on that cache, it works the same way as above till the information reach to the client. If the needed information is not cached on that cache, it will forward the request to the content server on the internet and also repeat the same steps as above till the information reached to the client.

In case of the reverse proxy, the same steps above are repeated except the request will forward by reverse way as in the forward proxy. Here, the request will forward from national level cache then to then to regional then to institutional bottom and finally to the content server in the internal network. The important note in caching hierarchy either in case of the forward proxy or the reverse proxy is each cache receives information from another level cache will cache a copy from that information for future need to the same request.

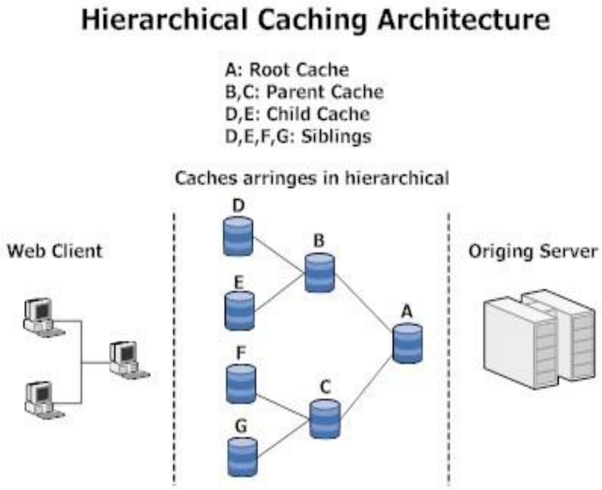


Fig. 3.1: Hierarichal caching architecture

3.2 Distributed Caching Architecture

In distributed Web caching systems, there are no other intermediate cache levels than the institutional caches, which serve each others' misses. In order to decide from which institutional cache to retrieve a miss document, all institutional caches keep meta-data information about the content of every other institutional cache. With distributed caching, most of the traffic flows through low network levels, which are less congested. In addition, distributed caching allows better load sharing and are more fault tolerant. Nevertheless, a large-scale deployment of distributed caching may encounter several problems such as high connection times, higher bandwidth usage, administrative issues, etc. [22]

There are several approaches to the distributed caching. Internet Cache Protocol (ICP), which supports discovery and retrieval of documents from neighboring caches as well as parent caches. Another approach to distributed caching is the Cache Array Routing protocol (CARP), which divides the URL-space among an array of loosely coupled caches and lets each cache store only the documents whose URL are hashed to it. [23]

3.3 Hybrid Caching

A hybrid cache scheme is any scheme that combines the benefits of both hierarchical and distributed caching architectures. Caches at the same level can cooperate together as well as with higher-level caches using the concept of distributed caching. [24]

A hybrid caching architecture may include cooperation between the architecture's components at some level. Some researchers explored the area of cooperative web caches (proxies). Others studied the possibility of exploiting client caches and allowing them to share their cached data.

One study addressed the neglect of a certain class of clients in researches done to improve Peer-to-Peer storage infrastructure for clients with high-bandwidth and low latency connectivity. It also examines a client-side technique to reduce the required bandwidth needed to retrieve files by users with low-bandwidth. Simulations done by this research group has proved that this approach can reduce the read and write latency of files up to 80% compared to other techniques used by other systems. This technique has been implemented in the OceanStore prototype (Eaton et al., 2004). [25]

IV. Design Goals & Proposed Architecture

To improve the computer network performance, decrease the workload for data center and ensure continual service improvement, we aim to design efficient mechanisms for reducing the workload of a data center and business Continuity verification and achieve the following goals:

- Reduces network bandwidth usage consumption which leads to reduce network traffic and network congestion
- Decrease the number of messages that enter the network by satisfying requests before they reach the server.
- Reduces loads on the origin servers.
- Decreases user perceived latency
- Reduced page construction times during both normal loading and peak loading
- If the remote server is not available due to a server \crash" or network partitioning, the client can obtain a cached copy at the proxy.

4.1 Proposed Architecture

We define before two types of proxies, the forward proxy and the reverse proxy. The forward proxy is used to forward clients from the clients on the internal network to the content server in the internet. The reverse proxy is used to forward requests from the clients in the internet to the content server in the internal network. Fig 5-1 shows that the forward proxy serves as a servant for internal clients and as a cache because it cached the content received from the content server on the internet. So for any the same repeated request, the forward server can return the cached content on it to the client directly without backing again to the content server. On the same time the forward proxy does an important rule as it hides the internal clients from outside world as the request is initiated from the forward proxy.

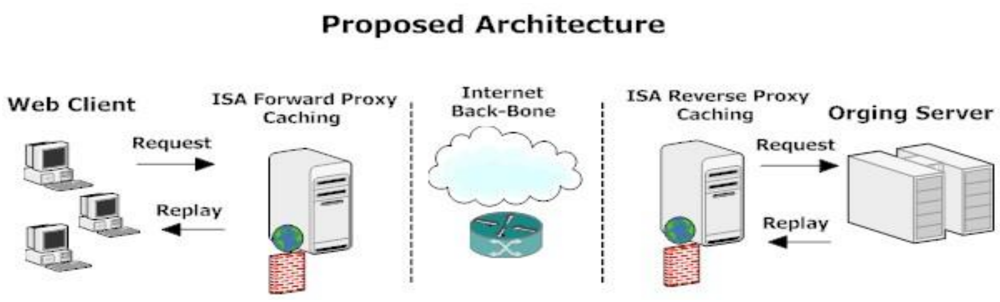


Fig. 4.1: Proposed Architecture

Fig 4-1 also shows the reverse proxy that used to forward the requests from external clients to content servers in internal network. In this case the reverse proxy makes encrypting content, compressing content, reducing the load on content servers. It also hides the responses from internal networks and as them come from the reverse proxy which increases the security. It also caches the content and forwards it directly to clients if they repeated again without backing again to the content server. Finally, we can use load balancing to balance between content servers and in this case the reverse proxy and forward the request from the client to any of this content serves which increase the availability of the system.

4.2 Proposed Architecture Workflow

1 The Remote Site client sends a request for Web Application content to the Forward proxy cache. If Forward proxy caching contains a valid version of the Web Application content in its cache, it will return the content to the requesting user.

2 If the content requested by Remote Site user is not contained in the Forward proxy cache, the request is forwarded to an upstream Reverse proxy caching.

3 If the upstream Reverse Proxy Cache has a valid copy of the requested content in cache, the content is returned to Forward proxy cache (Remote Site). Forward proxy cache places the content in its own cache and then returns the content to the Remote Site user who requested the content.

4 If the upstream Reverse proxy caching does not contain the requested content in its cache, it will forward the request to the Web Application server.

5 The Web Application server returns the requested content to reverse proxy caching. Reverse proxy caching places the content in cache.

6 Web Application server returns the content to reverse proxy caching. Reverse proxy caching server places the content in its cache. Reverse proxy caching server returns the content from its cache to Forward proxy. Forward proxy cache places the content in its own cache and then returns the content to the Remote Site user who requested the content.

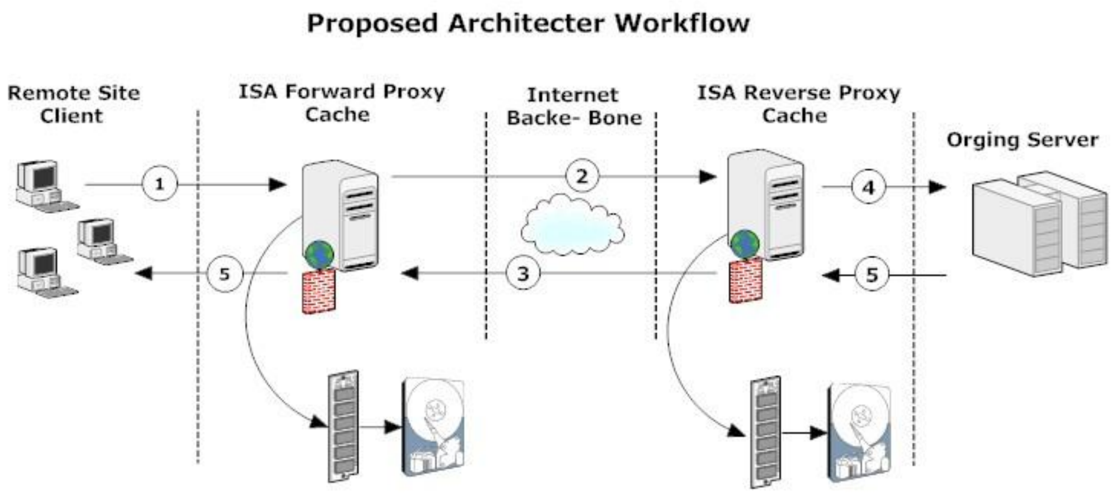


Fig. 4.2: Proposed workflow

V. Performance Analysis

In this chapter, we evaluate cache performance of web proxy caching for web applications and compare it to the case of not using web proxy caching at all. We will monitor and evaluate the performance of web proxy caching in three cases:

- Without using web proxy caching.
- At the beginning of using web proxy caching.
- After certain period (one month) from using web proxy caching.

We will take in our consideration the following parameters in evaluation process

- Requests returned from the application server.
- Requests returned from cache without verification.
- Requests returned from the application server, updating a file in cache.
- Requests returned from cache after verifying that they have not changed.

5.1 Performance Metrics

The seven main categories of performance metrics are:

1. ***Cache Performance:*** how requested Web objects were returned from the Web Server cache or from the network. It will be measured the according to
2. ***Traffic:*** the amount of network traffic, by date, sent through Web Proxy including both Web and non-Web traffic.
3. ***Daily traffic*****:** average network traffic through Web Proxy at various times during the day. This report includes both Web and non-Web traffic.
4. ***Web Application Responses:*** how ISA Server responded to HTTP requests during the report period.
5. ***Failures communicating:*** Web proxy Cache encountered the following failures communicating with other computers during the report period.
6. ***Dropped Packets:*** shows the users who had the highest number of dropped network packets during the report period Users that had the most dropped packets are listed first
7. ***Queue Length:*** The System\Processor Queue Length counter shows how many threads are ready in the processor queue, but not currently able to use the processor.

VI. Results

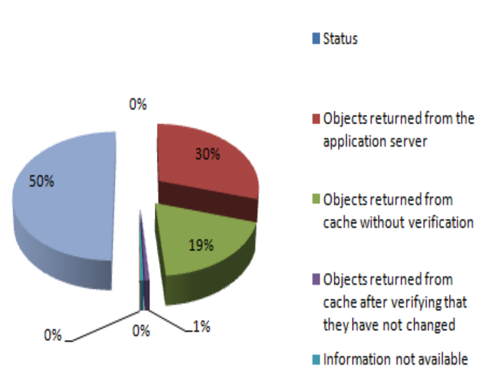
In this section we will investigate the performance analysis of cache, Network traffic, Failure communication, Dropped packets and queue length

6.1 Cache Performance

The cache performance results for each of the log files are shown below. The percentage of requests returned from cache without verification is high. It shows that between 38% of all requests result in a request returned from cache without verification, which is consistent with previously published results. Wills and Mikhailov[26] reported that only 15% to 32% of their proxy logs result in requests returned from cache without verification. Yin, et al.[27] revealed that 20% of requests to the server are due to revalidation of cached documents that have not changed. These results are consistent with the results found in our logs as discussed before. However, with the current logs, the number of requests returned from cache without verification has increased a little. This may be due to the duration of the analysis being longer for this study or to the use of different logs. It is assumed that a large fraction of these frivolous requests are due to embedded objects that do not change often.

Table 0-1: Cache Performance Results

Status	Requests	% of Total Requests	Total Bytes
Objects returned from the application server	20873	59.30 %	617.73 MB
Objects returned from cache without verification	13450	38.20 %	26.93 MB
Objects returned from cache after verifying that they have not changed	489	1.40 %	0.99 MB
Information not available	354	1.00 %	49.74 KB
Objects returned from the application server, updating a file in cache	59	0.20 %	14.62 KB
Total	**35225**	**100.00 %**	**645.72 MB**



6.2 Traffic

The results for average network traffic through web proxy caching Server at various times during the day at the beginning of using web proxy caching and after certain time from using web proxy caching are in Table below. The results indicate that the average processing time for handling the request is reduced by 43% after certain time of using web proxy caching because proxy caches the previous visited pages and return them directly to client without waste time to ask application server each time.

To reflect the physical environment of the network, we have to consider factors influencing traffic. Of various factors influencing traffic, object size is a factor of the objects themselves. Hence, we can reflect the size factor of web object. An average object size hit ratio reflects the factor of object size to object-hit ratio. Average-object Hit Ratio: The cache-hit ratio can indicate an object-hit ratio in web caching. The average object-hit ratio can calculate an average value of an object-hit ratio on a requested page the performance is evaluated by comparing an average object-hit ratio and response time [28] Response Time gain factor (RTGF): This factor give you amount of advantage in web cache response time. [29]

$$\text{Response Time Gain Factor} = \frac{(\text{Time Without cache} - \text{Time With cache}) \times 100}{\text{Time Without cache}}$$

Table 6-2: Average Network Traffic

At the beginning of using Web Proxy Cache				After certain time of using Web Proxy Cache				RTGF
Requests	Average Processing Time	Total Bytes	Cache Hit Ratio	Requests	Average Processing Time	Total Bytes	Cache Hit Ratio	
1924	133.00 sec	3.61 GB	0.00 %	6297	58.90 sec	2.94 GB	1.00 %	55.7142
1708	126.40 sec	9.75 MB	0.00 %	6157	61.30 sec	37.91 MB	0.00 %	51.5031
1654	127.10 sec	8.33 MB	0.00 %	5954	62.30 sec	36.27 MB	0.00 %	50.9834
1927	115.20 sec	8.42 MB	0.00 %	6004	58.00 sec	36.19 MB	0.00 %	49.6527
1731	121.10 sec	8.41 MB	0.00 %	5934	57.90 sec	36.12 MB	0.00 %	70.1156
1713	121.60 sec	8.51 MB	0.20 %	5901	58.30 sec	35.80 MB	0.00 %	50.1187
1642	126.60 sec	8.36 MB	0.00 %	5899	57.30 sec	35.80 MB	0.00 %	54.6318
1835	128.70 sec	8.48 MB	0.00 %	6249	54.50 sec	38.34 MB	2.80 %	57.6543
1859	126.30 sec	9.13 MB	0.00 %	6014	57.60 sec	36.69 MB	0.30 %	54.3942
2455	135.10 sec	11.50 MB	0.00 %	7212	52.00 sec	40.92 MB	12.00 %	69.7113

6.3 Traffic by Time of Day

The following Table (6.3) summarizes average network traffic through web proxy caching Server at various times during the day.

Table 6-3: Traffic by Time of day

Time Interval	Average Requests Per Second	Average Bytes Per Second	Average Response T ime for Cached Requests	Average Response Time for Non Cached Requests
00:00	15.4	67.49 KB	-	55.30 sec
00:15	17.1	13.15 MB	-	56.90 sec
00:30	14.8	10.57 MB	0.00 sec	
00:45	14.1	83.45 KB	-	65.20 sec
01:00	16.1	75.27 KB	0.00 sec	
01:15	16.2	69.21 KB	0.00 sec	
01:30	14.5	92.42 KB	0.00 sec	
01:45	14.0	77.47 KB	-	59.20 sec
02:00	15.8	67.94 KB	-	55.10 sec
02:15	16.9	68.87 KB	-	56.80 sec
02:30	15.2	93.02 KB	0.00 sec	
02:45	14.3	77.61 KB	-	52.60 sec
03:00	16.1	70.17 KB	0.00 sec	
03:15	16.7	70.64 KB	0.00 sec	
03:30	15.2	93.35 KB	0.00 sec	
03:45	14.5	77.47 KB	-	51.90 sec
04:00	15.9	70.57 KB	0.00 sec	
00:00	15.4	67.49 KB	-	55.30 sec
00:15	17.1	13.15 MB	-	56.90 sec
00:30	14.8	10.57 MB	0.00 sec	

As we shown in 6.3 Table the average response time for non cached requests (returned from origin servers) 55.30 sec for 67.49 KB in the first item, but the average response time for cached requests was it 0.00 sec for 10.57 MB Which shows that the use web proxy cache has Significant impact in the speed of retrieval of contents

Table 6-4: Web Application Responses

At the beginning of using Web Proxy Cache		
Response	**Requests**	**HTTP Responses**
Success	30910	87.80 %
Authorization failure	2565	7.30 %
Other	1094	3.10 %
Object moved	656	1.90 %
Object not found	0	0.00 %
Total	**35225**	100.00 %
After certain time of using Web Proxy Cache		
Response	**Requests**	**HTTP Responses**
Success	43518	91.60 %
Authorization failure	2403	5.10 %
Other	976	2.10 %
Object moved	622	1.30 %
Object not found	0	0.00 %
Total	**47519**	100.0

6.4 Web Application Responses

The results for how web proxy caching server responded to HTTP requests during the report period at the beginning of using web proxy caching server and after certain time from using web proxy caching. The results indicate that the percentage of success is increased.

6.5 Failures Communication

The results for how web proxy server encountered the following failures communicating with other computers during the report period. The results indicate that the percentage of errors after using web proxy caching is enhanced by noticeable value

Table 0-5: Failures Communication

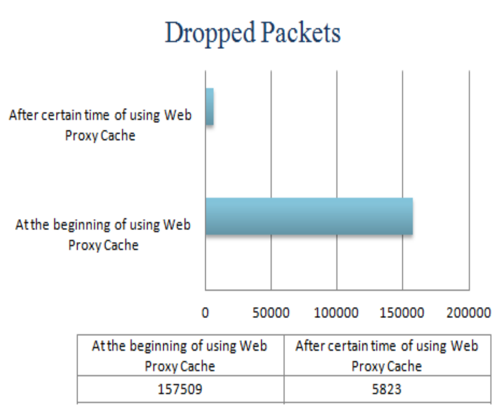
At the beginning of using Web Proxy Cache		
Error Type	**Errors**	**% Total Errors**
Firewall Service errors	165263	95.50 %
Web Proxy errors	7821	4.50 %
Total	**173084**	100.00 %
After certain time of using Web Proxy Cache		
Error Type	**Errors**	**% Total Errors**
Firewall Service errors	13062	58.70 %
Web Proxy errors	9184	41.30 %
Total	**22246**	100.0

6.6 Dropped Packets

The result below shows the users who had the highest number of dropped network packets during the report period. Users that had the most dropped packets are listed first. We can observe that the percentage of dropped packets is reduced by the time.

Table 6-6: Dropped Packets

At the beginning of using Web Proxy Cache			After certain time of using Web Proxy Cache		
User	**Dropped Packets**	**% of Total Dropped Packets**	**User**	**Dropped Packets**	**% of Total Dropped Packets**
10.1.12.13	**36857**	23.40 %	10.1.12.13	**692**	11.90 %
10.1.12.14	**35573**	22.60 %	10.1.12.14	**642**	11.00 %
172.31.1.2	**33909**	21.50 %	172.31.1.2	**183**	3.10 %
172.31.1.1	**31718**	20.10 %	172.31.1.1	**203**	3.50 %
10.1.12.12	**4752**	3.00 %	10.1.12.12	**794**	13.60 %
10.1.12.20	**2212**	1.40 %	10.1.12.20	**233**	4.00 %
10.1.12.23	**2211**	1.40 %	10.1.12.23	**211**	3.60 %
10.1.12.22	**2188**	1.40 %	10.1.12.22	**224**	3.80 %
10.1.12.15	**1536**	1.00 %	10.1.12.15	**565**	9.70 %
10.1.19.12	**1028**	0.70 %	10.1.19.12	**399**	6.90 %
10.1.10.17	**315**	0.20 %	10.1.10.17	**60**	1.00 %
Total	**152299**	100.00 %	**Total**	**7612**	100.00 %



6.7 Queue Length

After analyzing the network traffic, we found that network traffic is very burst over a large range of timescales and shows asymptotic self-similarity and multi-fractal behavior at intermediate time scales. Our study of request process of web-server shows that the busy-period traffic is very similar to a Harm parameter of around 0.7. [30] Also, several studies show that WAN traffic is multifractal in nature. We captured a sample from the state of network traffic of Giza and a Luxor station during the problem is occurring. Figure (6.7, 6.8) shows that Giza line Speed: 2048.0 kbits/s and Incoming packets 1990.9 kb/s (97.2%) ,Out packets 2001.3 kb/s (97.7%) and Louxur Line Speed: 2048.0 kbits/s with Incoming 1931.3 kb/s (94.3%) and Out packets 1931.5 kb/s (94.3%) ,This means that the network overloading might result in the loss of data packets and queuing delay

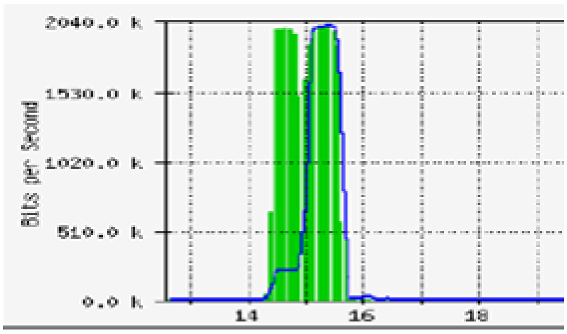


Fig. 0-7: Traffic Analysis for Giza-Station

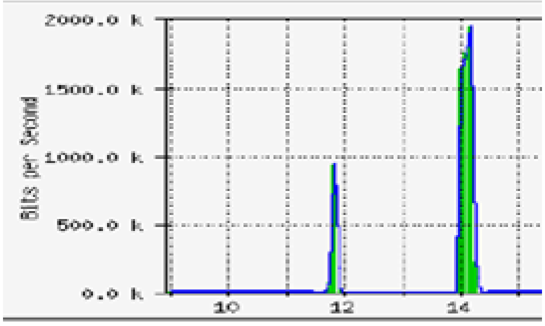


Fig. 6-8: Traffic Analysis for Luxour -Station

In Figure (6.7), (6.8) the multifractal behavior shows that there is a lot of requests arrive at the server [31] And any new arrival process will cause heavy-tailed queue-length distribution and as a result will lead to more loads on servers and the server cannot able to process any new request and return error to client this error “server-too-busy”. Due to the above problem, a multiple complaint from the system administrators for the origin servers loaded most of the time, and always those problems at certain times when the citizens buying tickets. also The widely use of web-based user interfaces for conducting business on the ENR network has brought to focus the problem of dealing with overloads to which the web servers (especially those that form the front end servers).

Table 6-7: CPU Utilization and Avg Queue Length

At the beginning of using Web Proxy Cache			After certain time of using Web Proxy Cache		
Overload (requests/s)	**Processor Utilization %**	**Avg Queue Length** (Reqs)	**Overload** (requests/s)	**Processor Utilization %**	**Avg Queue Length** (Reqs)
36.857	80.95	37.98 %	6.92	692	0.97
35.225	73.67	47.54 %	6.42	642	1.02
30.910	74.67	19.02 %	1.83	183	0.21
31.718	79.67	66.41 %	2.03	203	0.37

VII. Conclusions

We proposed a hybrid of web proxy caching architecture to improve the performance of computer network, by integrating forward proxy caching technique and reverse proxy caching technique. The forward proxy is used to forward clients from the clients on the internal network to the content origin server. We are use the forward proxy caching to reduce the network traffic, reduced page construction times during both normal loading and peak loading , decreasing dropped packets. The reverse proxy is used to forward requests from the clients in the internet to the content server in the internal network. We are use the reverse proxy caching Reducing the traffic on the content servers in case the request is blocked by the reverse proxy. In this case the request is rejected directly by the reverse proxy without interrupt the content servers and reducing the bandwidth consumes by blocked requests as it is blocked directly by reverse proxy before reaching to the content servers. The forward proxy serves proxy is used as a servant for internal clients and as a cache because it cached the content received from the content on origin server. So for any the same repeated request, the forward server can return the cached content on it to the client directly without backing again to the content server. On the same time the forward proxy does an important rule as it hides the internal clients from outside world as the request is initiated from the forward proxy.

Web proxy caching is effective solution to lessen Web service bottleneck, reduce traffic over the network and improve scalability of the Web Application system. This research presents an overview of the Web Proxy caching Technique design and management of a cache consistency algorithm and examines the proposed solutions to improve computer networks performance in recent literature.

- We proposed an effective and flexible for ENR network by using web proxy caching technique to Reduces network bandwidth usage, Decrease the number of messages that enter the network, Reduces loads on the origin servers.
- We have designed and implemented our web proxy caching in a system involving two major components. A forward proxy cache for users, Reverse proxy cache for server's application.
- We have performed an analytical investigation of four proposed consistency algorithms to identify how read and writes could affect the operation of these algorithms and to provide taxonomy of when the algorithms are most appropriate.
- We evaluated our algorithm and compared it with existing and proposed algorithms using a testbed of cluster nodes to run an emulation of the Web. This included the used of actual Web proxies and servers that are driven by a workload of server and proxy logs.
- We investigated the amount of unnecessary traffic that is generated on the Web application today. This provides estimates of the possible reduction of unnecessary messages that could occur if our approach were available for use in the Web application today.
- Our background study determines whether significant network savings would occur if server invalidation played some part in the maintenance of consistency. These evaluations show the potential for substantial improvement offered by our proposal.
- A survey of existing techniques shows that many individual techniques are good in their own context, but we have shown in our results that an integration of these techniques performs significantly better when compared to the techniques in isolation.
- We implemented a web proxy-based content accelerator, which combines the best features of fragment - caching and proxy-caching, with the intention of reducing bandwidth requirement.

Acknowledgements

I would like to express our gratitude to Prof. DR/ Nashaat El- Khameesy Computers & Information systems Dept- Sadat Academy, Asmaa Ahmed - Communications and Electronic Dept- Engineering Faculty, Cairo University The authors thankfully acknowledge the support provided by Trans IT Datacenter Operation Team

References

[1] J. B. Pati , B. V. Pawar / Improving Performance on WWW using Intelligent Predictive Caching for Web Proxy Servers, IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 1, January 2011

[2] Chen, T.: Obtaining the optimal cache document replacement policy for the caching system of an EC website. European Journal of Operational Research 181(2), 828–841, 2007

[3] Romano, S., ElAarag, H.: A neural network proxy cache replacement strategy and its implementation in the Squid proxy server. Neural Computing & Applications 20(1), 59–78, 2011

[4] Cobb, J., ElAarag, H.: Web proxy cache replacement scheme based on back-propagation neural network. Journal of Systems and Software 81(9), 1539–1558 ,2008

[5] Kaya, C.C., Zhang, G., Tan, Y., Mookerjee, V.S.: An admission-control technique for delay reduction in proxy caching. Decision Support Systems 46(2), 594–603 , 2009

[6] Kumar, C., Norris, J.B.: A new approach for a proxy-level web caching mechanism. Decision Support Systems 46(1), 52–60 ,2008

[7] Kumar, C.: Performance evaluation for implementations of a network of proxy caches. Decision Support Systems 46(2), 492–500 ,2009

[8] Rajeev Tiwari et al. /Robust Distributed Web caching: A Dynamic clustering Approach , International Journal of Engineering Science and Technology (IJEST), Vol. 3 No. 2 Feb 2011

[9] Ali, W., Shamsuddin, S.M., Ismail, A.S.: Web proxy cache content classification based on support vector machine. Journal of Artificial Intelligence 4(1), 100–109 ,2011

[10] Tiwari Rajeev and Khan Gulista, "Load Balancing in Distributed Web Caching: A Novel Clustering Approach", Proc. of ICM2ST-10, International Conference on Methods and models in science and technology pp. 341-345, November 6, vol.1324, 2010

[11] Rajeev Tiwari, Gulista khan, "Load Balancing through distributed Web Caching with clusters", Proceeding of the CSNA 2010 Springer, pp 46-54, Chennai, India, 2010

[12] Takaaki Komura, Hiroaki Sano, Noritoshi Demizu, Ken Makimura, "Design and Implementation of Web Forward Proxy with Shibboleth

Authentication," saint, pp.321-326, 2011 IEEE/IPSJ International Symposium on Applications and the Internet, 2011

[13] Wang, Xiao-Ming Fu, Fang-Wei /Forward secure proxy signature scheme, Tongxin Xuebao (Journal on Communications). Vol. 26, no. 11, pp. 38-42. Nov. 2005

[14] P. Wurzinger,C. Platzer,C. Ludl,E. Kirda, C. Kruegel /SWAP: Mitigating XSS attacks using a reverse proxy, Pages 33-39 , IWSESS '09 Proceedings of ICSE Workshop on Software Engineering for Secure Systems,2009

[15] Lin, C.-H., Liu, J.-C., and Lien, C.-C., "Detection Method Based on Reverse Proxy against Web Flooding Attacks," 8th International Conference on Intelligent Systems Design and Applications (ISDA-2008), pp.281- 284. Kaohsiung City, Taiwan. (2008)

[16] Peter Sommerlad /Reverse Proxy Patterns, (c) 2003 by Peter Sommerlad,Erlenstrasse 79, CH-8832 Wollerau, Switzerland,2003

[17] C. Amza and G. Soundararajan and E. Cecchet. Transparent Caching with Strong Consistency in Dynamic ContentWeb Sites. In ICS, 2005.

[18] Laura Abba, Marina Buzzi, and Massimo Ianigro. Introducing transparent Web caching in a local area network. In Proc. of Intl. CMG Conference, pages 99–110, 2000.

[19] Wenting Tang, Fan Du, Matt W. Mutka, Lionel M. Ni, and Abdol-Hossein Esfahanian. Supporting global replicated services by a routing-metric-aware DNS. In Proceedings of the 2nd International Workshop on Advanced Issues of E-Commerce and Web-Based Information Systems (WECWIS 2000), pages 67–74, June 2000

[20] L Abba, M Buzzi, D Pobric, M Ianigro/ Introducing transparent web caching in a local area network, Centre National de la Recherche Scientifique Paris, France, France ©2000

[21] Rodriguez, P., Spanner, C., Biersack, and E., Analysis of Web Caching Architectures: Hierarchical and Distributed Caching, IEEE/ACM Transactions on Networking, Vol. 9(4):404-418, August 2001.

[22] S. V. Nagaraj, web caching and its applications, page 24, Caching Architectures, print ©2004 kluwer Academic Publishers- new york , boston-2004

[23] Suresha , Caching Techniques for Dynamic Web Servers, Doctor of Philosophy Department of Computer Science and Automation ,Indian Institute of Science Bangalore 560 012 INDIA - June 2007

[24] Jinsuk Baek,Gurpreet Kaur and Junghoon Yang /A New Hybrid Architecture for Cooperative Web Caching, journal of ubiquitous convergence and technology, VOL. 2, NO. 1, MAY 2008

[25] S. Rhea, P. Eaton, D. Geels, H. Weatherspoon, B. Zhao, and J. Kubiatowicz. Pond: the OceanStore prototype. In Proc. Of USENIX FAST, pages 1–14, Mar. 2003.

[26] C. E. Wills and M. Mikhailov. Toward a better understanding of web resources and server responses for improved caching. pages 1231-1243, 1999

[27] J. Yin, M. Dahlin, and A. Iyengar. Engineering server-driven consistency for large- scale dynamic web services. In Proceedings of the 10th International World-wide Web Conference, pages 46-57, Hong Kong, May 2001

[28] Yun Ji Na ,Seok KO2 / Design and performance Analysis of the Web Caching Algorithm for Performance Improvement of the Response Spee,. T.-J. Cham et al. (Eds.): MMM 2007, LNCS 4352, Part II, pp. 686 – 693, 2007. Springer-Verlag Berlin Heidelberg 2007

[29] Dhawaleswar rao. ch / study of the web caching algorithms for performance improvement of the response speed, Journal of Computer Science and Engineering (IJCSE) ,Vol. 3 No. 2 Apr-May 2012

[30] Lin, C.-H., Liu, J.-C., and Lien, C.-C., "Detection Method Based on Reverse Proxy against Web Flooding Attacks," 8th International Conference on Intelligent Systems Design and Applications (ISDA-2008), pp.281- 284. Kaohsiung City, Taiwan. (2008)

[31] Peter Sommerlad /Reverse Proxy Patterns, (c) 2003 by Peter Sommerlad,Erlenstrasse 79, CH-8832 Wollerau, Switzerland,2003

Authors' Profiles

Dr Nashaat el-Khameesy: Prof. and Head of Computers & Information systems Chair, Sadat Academy, Maady, Cairo-Egypt.

Hossam Abdel Rahman Mohamed: Master of computer science & information system - Sadat Academy - Dept Computer and Information System, Maady, Cairo, Egypt - He is currently position Data Center Operation Team leader at Trans IT Company - Ministry of Transportation