Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 4, Issue. 11, November 2015, pg.13 – 28

RESEARCH ARTICLE

Cost and Benefit Analysis of Integrating Forwarding and Caching Servers in Organization's Network Mainstream

Amadi E. C.[1], Ndanike Paul E.[2], Ukeoma Pamela E.[3], Osuagwu Vivien A.[4]

[1234]*Department of Information Management Technology, Federal University of Technology Owerri, Imo State, Nigeria*

[1] ec.amadi.server@gmail.com, [2] paulndanike@gmail.com, [3] pamelaukeoma@yahoo.com,

[4] osuagwuvivien@yahoo.com

Abstract -- Organizations (financial institutions, educational institutions etc) all over the globe carry out their day-to-day activities on the internet which to a very large extent has been one of their major driving forces towards rapid development. In a bid to sustain the growth, the need for an efficient and cost effective network cannot be over emphasized. The purpose of LAN based caching and forwarding is to improve the organization's network efficiency and reliability by reducing the amount of traffic between the LAN and the Internet. Beyond distributing load away from server "hot spots", caching and forwarding can also save bandwidth, reduce latency, and protect the network from clients that erroneously loop and generate repeated requests. This paper tends to analyze the cost and benefit of integrating forwarding and caching servers in organizations network mainstream.

Keywords: Forwarder or forwarding server, caching server, Domain Name System (DNS), IP Address

1.0 Introduction

Each time a server receives a query for a name that is not in its domain, it needs to search its database for a server IP address. Reduction of this search time would increase efficiency. DNS handles this with a mechanism known as **caching.** When a server asks for a mapping from another server and receives the response, it stores this information in its cache memory before sending it to the client. If the same or another client asks for the same mapping, it can check its cache memory and solve the problem.

A forwarder on the other hand is a Domain Name System (DNS) server on a network that is used to forward DNS queries for external DNS names to DNS servers outside that network. A DNS server on a network is designated as a forwarder when the other DNS servers in the network are configured to forward the queries that they cannot resolve locally to that DNS server. Forwarding servers or forwarders, process requests that DNS servers cannot resolve locally. A forwarding server which is not a separate type of server rather a caching server used in a particular way works in combination with caching servers. An effective Domain Name System (DNS) is critical to Internet access speeds. The bandwidth of your Internet connection is irrelevant if the DNS system is slow. DNS supports high performance, availability, and scalability through the use of data hierarchies, data replication, and caching.

2.0 Domain Name System (DNS)

DNS is an abbreviation for **Domain Name System**. This system was created to translate the domain names of servers into IP addresses. It works like a phone book as it connects your IP address (phone number) to a website (name).

This permits computers and networking devices a means of understanding one another. This information is stored by a DNS Server, which provides the proper information to the requesting computer or networking device. DNS can also provide auxiliary services like authentication, verification, and configuration details of specific records.

Without DNS, the internet wouldn't able to function. Rather, we would have to manually type in the IP address of every website we wanted to visit.

DNS uses a distributed database system that contains information on domains and hosts within those domains. The information is distributed among many name servers, each of which holds part of the database. The distribute nature of the system has the twin advantages that the maintenance of the system is delegated and the loss of one DNS server does not prevent name resolution from being performed.

3.0 Caching DNS Server

A caching server is not authoritative for any zone. Instead, it handles queries by asking other servers for information. This type of server is also known as a resolver because it handles recursive queries and generally can handle the grunt work of tracking down DNS data from other servers.

When a caching DNS server tracks down the answer to a client's query, it returns the answer to the client. But it also stores the answer in its cache for the period of time allowed by the records' Time to Live (TTL) value. The cache can then be used as a source for subsequent requests in order to speed up the total round-trip time.

Almost all DNS servers that you might have in your network configuration will be caching DNS servers. These make up for the lack of adequate DNS resolver libraries implemented on most client machines. A caching DNS server is a good choice for many situations. If you do not wish to rely on your ISPs DNS or other publicly available DNS servers, making your own caching server is a good choice. If it is in close physical proximity to the client machines, it is also very likely to improve the DNS query times.

3.1 Benefits of integrating caching servers in an organization’s network stream

The purpose of LAN based caching is to improve network efficiency by reducing the amount of traffic between the LAN and the Internet. The most obvious and most frequently cited benefit is the shorter time required for the caching server to deliver cached content. Delivery time and therefore the end user experience are enhanced dramatically. For example, delivery of a 100KB web page from the originating server to the end user would take about 17 seconds over a 56Kbps modem, or 7.8 seconds over a dual channel Multilink PPP ISDN connection, assuming that there was no additional traffic congestion at the ISP or on the Internet backbone (this is not necessarily the case). The same page would take one second to deliver over a T1 connection, again assuming perfect Internet traffic conditions. However, this same page would be delivered from the caching server to the end user in about one tenth of a second *regardless of Internet traffic conditions.* This is the first and most ***obvious benefit of caching***.

Some will point out that this is only a benefit if the same content is viewed a second time by a different user. If this does not occur, it may be argued, caching would be of no benefit.

However, repeat visits to the same Web sites are more frequent than one may think. A recent test2 representing a small business setup with a LAN comprising about 30 computers revealed that up to 70% of content delivered during any one hour period came from the cache, and as little as 30% came from the Internet. An independent laboratory study3 recently showed average response times reduced by 87% with the use of a caching server.

It is a commonly held belief among users of modem Internet connections that by upgrading to a cable modem or ADSL connection, all of their bandwidth worries will disappear. Nothing could be further from the truth. ***Let us consider what happens when a user clicks on a hyperlink or types a URL in their browser:***

1. The browser performs a DNS lookup which in and of itself can take several seconds.

2. The DNS lookup returns a numerical IP address that the browser requires in order to find the web page indicated by the hyperlink.

3. The browser then sends a request to its own gateway or router.

4. The gateway or router forwards the request to another router. This is repeated for as many routers as there are between the client and the server. When the request is forwarded, this is called a "hop". There can be up to ten or more hops in a connection. Internet congestion can cause the delay in a single hop to last for more than one second.

5. The server that is hosting the web page receives the request and responds by returning the object requested by the browser.

6. Step 4 is repeated in the opposite direction.

7. The browser receives the object, which is essentially a list of all of the other objects that make up the web page. It then goes through the list of items comprising the web page, and for each item (there may be 10, 20, 30 or more), steps 3 to 6 are repeated, including any delays between hops. Steps 1 and 2 may also be repeated for items to be retrieved from different servers.

Let's consider what happens when a user requests cached content from a LAN based caching server:

1. The browser performs a DNS lookup which in many cases is managed through DNS caching and takes a fraction of a second.

2. The browser sends the request to the caching server.

3. The caching server immediately returns the list of page items with no hops and no delays.

4. The browser goes through the list and requests all of the items from the caching server, which are in turn delivered immediately at LAN speeds.

Many servers deliberately throttle back their throughput to prevent users from dominating their bandwidth throughput when viewing cumbersome graphics.

This contributes to additional delays, and happens most frequently on sites with pages containing large elements. The larger the elements, and the more of them there are, the greater the benefit of caching. Thus, the performance gains experienced by users of web caching servers are potentially as substantial for T1 connections as they are for modem connections.

In our calculations we have not yet taken into account the distance travelled by the information.

If the server is thousands of miles away and the page requested is made up of many components, the accumulated distance travelled by all of the requests and responses may add up to hundreds of thousands of miles. Signals travel at approximately 60% of the speed of light.

Even with eutopic bandwidth and traffic conditions, it may take several seconds for content to travel such distances as opposed to near instantaneous LAN delivery.

Whatever the type of Internet connection, there are benefits to optimizing bandwidth usage, particularly as Internet content volumes increase.

Caching DNS server can significantly provide a faster network speed and also reduce the amount of data downloaded from the internet. Think of it this way: It's Monday morning and Microsoft has released a new OS update that is 500MB in size. If you have 50 clients and each one of those devices has to connect to Microsoft's servers over the Internet to download the update, that's 25GB of data downloaded directly from Microsoft using a connection that is much slower than your private network. If you're using a caching server, 500MB gets downloaded from Microsoft and the other 24.5GB are downloaded from your server, making the most of your faster network speeds and reducing the amount data downloaded from the Internet.

There are additional benefits as well. By delivering content from its own cache, the caching server reduces bandwidth use between the LAN and the Internet. This means that more bandwidth becomes available for users requesting fresh content directly from the Internet.
These users experience improved response times even if they request content that is not stored in the cache.
Ironically, there are cases where a browser may display web content from the LAN based web cache faster than from its own disk cache. Because browsers are optimized for content delivered over the network, some may actually display a page delivered over the LAN more quickly than if the same page were read from their own computer's disk.

However, in order to gain the most benefit from a Caching DNS Server, you must not reboot the computer. Since the DNS Cache only remains in RAM (or sometimes on disk in the page file), the contents of the cache will be lost if the server is rebooted, Be sure to include fault-tolerance mechanisms such as an UPS, Disk Mirroring, and redundant power supplies on such a machine.

4.0 Forwarding Server

A forwarder or forwarding server is a Domain Name System (DNS) server on a network that is used to forward DNS queries for external DNS names to DNS servers outside that network. You can also configure your server to forward queries according to specific domain names using conditional forwarders. A DNS server on a network is designated as a forwarder when the other DNS servers in the network are configured to forward the queries that they cannot resolve locally to that DNS server. By using a forwarder, you can manage name resolution for names outside your network, such as names on the Internet, which can improve the efficiency of name resolution for the computers in your network.

Forwarding server is not a separate type of server, rather a caching server used in a particular way. A forwarding DNS server will look almost identical to a caching server from a client's perspective, but the mechanisms and work load are quite different.

In organizations where large number of caching servers is used, forwarding servers are found useful as they tend to reduce traffic on their connection to the internet. If the caching servers in the organization’s network cached names and IP address independently, the information would be duplicated because each caching server would access the internet to resolve names. A forwarding server acts as the single point of contact for the internet.

When you designate a DNS server as a forwarder, you make that forwarder responsible for handling external traffic, which limits DNS server exposure to the Internet. A forwarder builds up a large cache of external DNS information because all the external DNS queries in the network are resolved through it. In a small amount of time, a forwarder resolves a large number of external DNS queries using this cached data. This decreases the Internet traffic over the network and the response time for DNS clients.

A DNS server that is configured to use a forwarder behaves differently than a DNS server that is not configured to use a forwarder. *A DNS server that is configured to use a forwarder behaves as follows:*

1. When the DNS server receives a query, it attempts to resolve this query by using the zones that it hosts and by using its cache.

2. If the query cannot be resolved using local data, the DNS server forwards the query to the DNS server that is designated as a forwarder.

3. If forwarders are unavailable, the DNS server attempts to use its root hints to resolve the query.

When a DNS server forwards a query to a forwarder, it sends a recursive query to the forwarder. This is different than the iterative query that a DNS server sends to another DNS server during standard name resolution (name resolution that does not involve a forwarder).

4.1 Benefits of integrating forwarding servers in an organization's network stream

Forwarders are often desirable when access to remote DNS servers requires use of a slow link, such as a fast speed internal network linked to the Internet over a relatively low-speed connection. Using forwarders in this situation can cut down on expensive traffic over the low-speed link in two ways:

1. ***Reduce the number of overall queries that get sent across the slow link*** — for example, if your DNS server has a slow, expensive dial-up connection to an Internet service provider (ISP). When the DNS server used as a forwarder for your internal network receives a query for a remote name on the Internet, it can directly contact remote servers on the Internet. It can repeat additional queries until it determines the authoritative server for the name. Once it finds the authoritative server, the forwarder contacts it and receives a complete response. Another option that can reduce traffic is using a DNS server on the Internet as a forwarder..
2. ***Share remote results on your local network:*** Forwarders provide a way to share information about remote names with a group of DNS servers located in the same area. For example, assume your organization has several DNS servers on a LAN. Rather than having each server send queries through a firewall and out to the Internet, all DNS servers are configured to forward queries to one DNS server (perhaps sitting on the firewall) that makes the necessary queries to the remote servers. In the process, the forwarder builds up a cache of Internet DNS names from the responses it receives. Over time, as local DNS servers continue to forward queries to it, the forwarder answers more queries from its cache because it begins to have an increasing number of answers based on previous queries for the same or similar names.

5.0 Integrating Caching and Forwarding Servers in a Network

Forwarding servers work in combination with caching servers. The figure 5.1 below explains the operations of a caching and forwarding servers in a particular network.

a) Workstation 1 enters www.microsoft.com into a browser
b) Caching server 1 asks forwarding server to resolve it
c) 3. Forwarding server resolves it and caches (saves) it
d) Caching server 1 caches it as well

e) The IP address is returned to workstation 1 and the page is requested
f) Workstation 2 enters www.microsoft.com into a browser
g) It is automatically resolved at caching server 1
h) Workstation 3 enters www.microsoft.com into a browser
i) Caching server 2 cannot resolve it but the forwarding server can

Without the forwarding server, the caching server would have to access the internet. In an event where all the caching servers cached names and IP addresses independently, this will give rise to duplication of information because each caching server has to access the internet to resolve names. As a result, a forwarding server acts as the single point of contact to the internet, so it caches frequently-used names requested by other servers. For instance, peradventure the users of five caching servers wanted to access the www.futo.edu.ng, and then each caching server would have to go to the internet so as to resolve the address. If the caching servers used a forwarding server, the story would not be the same as the forwarding server would get the IP address for www.futo.edu.ng and cache it. Whenever the caching servers request for the IP address of www.futo.edu.ng, they could get it from the cache at the forwarding server. **Workstation in Organization**

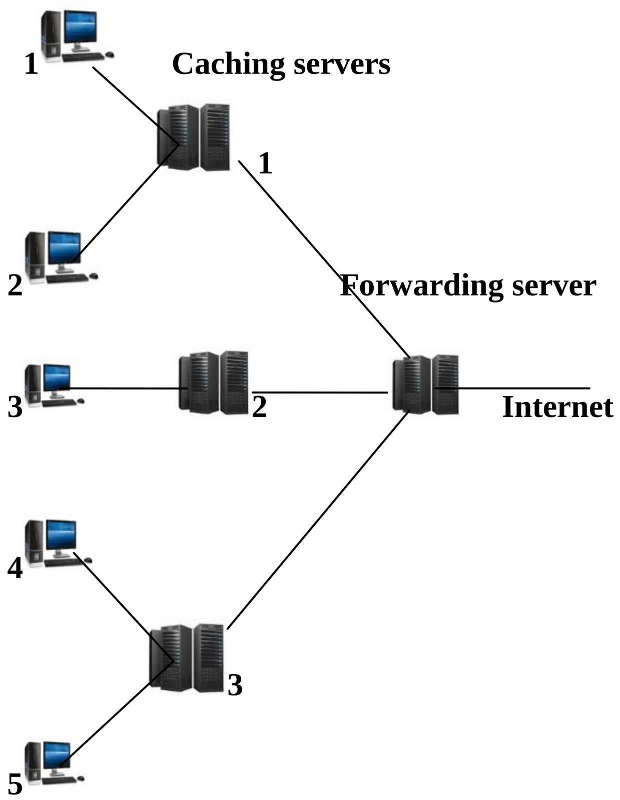


Figure 5.1: caching server using a forwarding server

6.0 Cost of acquiring a Server?

We will look at the three major components to a server's overall costs, which include:
- Cost of hardware
- Cost of server operating system and applications
- Cost to administer

Cost of Server Hardware

The tasks performed by a server as well as the number of users it's expected to serve will largely determine your server's hardware costs. While servers are almost always more expensive than their desktop counterparts, those on a tight budget can find low-cost server options that handle many if not all of the tasks your business will likely need from a server.
If your server will primarily be used for tasks like print serving and office document file sharing among fewer than 25 users, a server with a low-end processor, as little as 1GB to 2GB of RAM, and 500GB to 1TB of RAID storage will most likely suffice, and should cost your business as little as N80,000.
On the other end of the scale are high-end servers for tasks like data-intensive Web and database serving, video storage and sharing and enterprise-grade messaging and conferencing. These servers will typically include multiple processors, 16GB or more of RAM, expansive storage with multiple redundancies, and a high-end server operating system, and they can cost in the thousands or tens of thousands of dollars. High-end servers may also be deployed in multiple server configurations known as clustering.

Cost of Server OS and Apps

The server you purchase may or may not include a server operating system. When it comes to selecting your own server operating system, high-end server OSes like Windows Server 2008 R2 start at about N200, 000 for up to five users (five client access licenses, or CALs) and N80, 000 for the Enterprise edition, which includes 25 CALs. Mac OS X Server, N99,800 provides an unlimited client licenses to numerous clients.
Microsoft Small Business Server (approximately N110,000 with support for up to 25 users) and a variety of enterprise Linux server distributions that range from free to N200,000 or more for an annual support subscription.
You'll also need to budget for the software applications your server will need in order to perform its tasks. The dollar amounts can add up quite quickly in this area – for example, to handle e-mail services you'll need an application like Microsoft Exchange Server, which starts at N139,800 for the
Standard edition and N799, 800 for the Enterprise edition; for database services an app like Microsoft
SQL Server will cost nearly N180, 000; and for file sharing and online collaboration, options like
Microsoft's SharePoint Server or Citrix Presentation Server can cost anywhere from N800, 000 to N600, 000 or more.

Cost to Administer a Server

In most cases, a server's hardware and software costs represent only a small part of the total cost of ownership for a server. In fact, the hardware and software costs typically account for only 15 to 25 percent of the overall costs associated with installing, maintaining, upgrading and supporting a dedicated server.
For many businesses, the server becomes the lifeline to its success and any downtime can be disastrous, making it critically important to invest in your server's ongoing operation. As a result, when budgeting for a server, it's imperative to create a solid plan for the costs associated with configuring and administering your server – costs that include initial configuration and ongoing support fees, workforce costs for day-to-day administration, reserve funds for replacing hardware as warranties expire, software update fees and more.

7.0 Cost analysis of integrating caching and forwarding servers in an organization's network stream

Cost savings: It is obvious that the time spent waiting for web pages to load is not used productively. If this waiting time is reduced, there are demonstrable cost savings to the organization.

In addition, the bandwidth economy realized through the use of a caching server and forwarding server will in many cases allow a business to avoid upgrading to a higher bandwidth internet connection. Cost savings achieved in this way can be very considerable.

Equipment costs which includes networking related equipments and servers; with regard to capital expenditure, considering the prices of many components (servers) that will be involved will be greater in number and therefore there will be an increase in the total costs for equipments.

Power costs; In summing up the power consumption of each type of components (servers) used, we compute power consumption costs by adding up the power consumption due to each of the components, which shows an increase in cost.

Cabling Cost

We compute cabling costs same as above

Overall cost and discussion

We now consider the total cost as the sum of equipment, cabling and power costs etc.

Note that we do not account for other types of cost such as **management costs,** which are difficult to quantify.

8.0 Cost function analysis of the web hierarchical caching model

Web proxy caching hierarchy is one of the main solutions used to improve Internet QoS. Focusing on the modeling of Web proxy hierarchy caching, this paper applies cost function to analyze Web caching performance.

In Web caching hierarchical model, if the requested object has been already in cache, the cache is able to satisfy request, resulting in a cache hit.

Otherwise, the cache miss occurs.

Table 1: Symbol definitions

Notation	Description
S_t	The size of object t
Q_t	Q_t = 1, the cache hit Q_t = 0, the cache miss
N	The total number of Requests originated at the clients
B	To total bytes of N requests
HR_j	HR at j-th level(1≤j≤m)
BHR_i	BHR at j-th level(1≤j≤m)

If the request is not found at any cache level, it will be send to the Web server. Mostly Web evaluation performance metrics are captured by HR and, Besides, the symbols of Eq. 1 and 2 are described in Table 1.

HR (Hit Rate) is the number of requests satisfied by cache divided by the N cacheable requests seen by the cache.

$$HR = \frac{\sum_{t=1}^{N} Q_t}{N} \text{ X } 100\% \qquad (1)$$

BHR (Byte Hit Rate) emphasizes the total bytes saved by caching certain objects.

$$BHR = \frac{\sum_{t=1}^{N} Q_t \, x \, S_t}{\sum_{t=1}^{N} S_t} \text{ X } 100\% \qquad (2)$$

If the request cannot be retrieved from lower level cache, it will be directly sent to the upper level cache. In this case, it will cause data transfer or delay.

Here, assume that each cache has the same caching capacity. C1 indicates the cost which is generated by the request being sent to the first-level cache

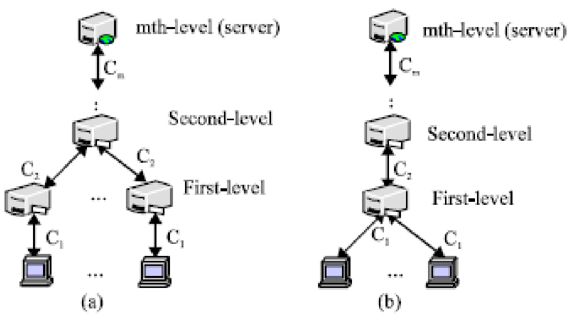


Fig. 1: Cost analysis of m-level hierarchical caching model

In addition to C1, define Cj (2≤j≤m) as the cost associated with two adjacent levels. The requests are sent to the fist-level cache with the same probability, so the first-level cache can provide similar HR or BHR. In Fig. 1a, the total number of requests N1 satisfied at the first-level cache is:

$$N_1 = \sum_{i=1}^{m}(HR_1 \times \frac{N}{n}) = HR_1 \times N$$

Similarly, in Fig. 1b, the total number of requests satisfied at the first-level cache is: N1 = HR1xN, we can conclude N1 = N1. It indicates that the Fig. 1b can take place of Fig. 1a for the cost analysis in Web hierarchical caching model. Nj represents the total number of requests satisfied at the jth-level cache and RCj signifies the cost of satisfied Nj requests. They can be respectively defined as:

$$N_j = HR_j \times \ (N - \sum_{i=1}^{j=1} N_i) \qquad (3)$$

$$RC_j = (\sum_{i=1}^{j} C_i) \times N_j \qquad (4)$$

Bj represents the total requested bytes at the jth-level cache. BCj means the cost of satisfied Bj bytes. Both of them can be calculated as Eq. 5 and 6.

$$B_j = BHR_j \times \ (B - \sum_{i=1}^{j=1} B_i) \qquad (5)$$

$$BC_j = (\sum_{i=1}^{j} C_i) \times B_j \qquad (6)$$

The total cost of requests and bytes served from the caches can be expressed as RC and BC.

$$RC = RC_1 + RC_2 + \ldots + RC_m = \sum_{j=1}^{m}\{(\sum_{j=1}^{j} C_i) \times N_j\} \qquad (7)$$

$$BC = BC_1 + BC_2 + \ldots + BC_m = \sum_{j=1}^{m}\{(\sum_{j=1}^{j} C_i) \times B_j\} \qquad (8)$$

Supposing all the requests hit at the server, the total cost of requests and bytes can be expressed as RCworst and Bcworst

$$RC_{worst} = RC_1 + RC_2 + \ldots + RC_m = (\sum_{j=1}^{j} C_i) \times N \qquad (9)$$

$$BC_{worst} = BC_1 + BC_2 + \ldots + BRC_m = (\sum_{j=1}^{j} C_i) \times B \qquad (10)$$

In general, the requests hit at some cache or server. On this occasion, the cost of total requests and bytes can be expressed as RRC and RBC.

$$RRC = \frac{RC}{RC_{worst}} = \frac{\sum_{i=1}^{j}\{(\sum_{i=1}^{j} C_i \times N_j\}}{(\sum_{j=1}^{j} C_i) \times N} \quad (11)$$

$$RBC = \frac{BC}{BC_{worst}} = \frac{\sum_{i=1}^{j}\{(\sum_{i=1}^{j} C_i \times B_j\}}{(\sum_{j=1}^{j} C_i) \times B} \quad (12)$$

In order to facilitate the cost analysis, assuming Cj = C (1≤j≤m). According to Eq. 11 and 12, we can deduce Eq. 13 and 14.

$$RRC = \frac{N_1 + 2N_2 + \ldots mN_m}{mN} \quad (13)$$

$$RRC = \frac{B_1 + 2B_2 + \ldots mB_m}{mB} \quad (14)$$

Analyzing the following two cases:

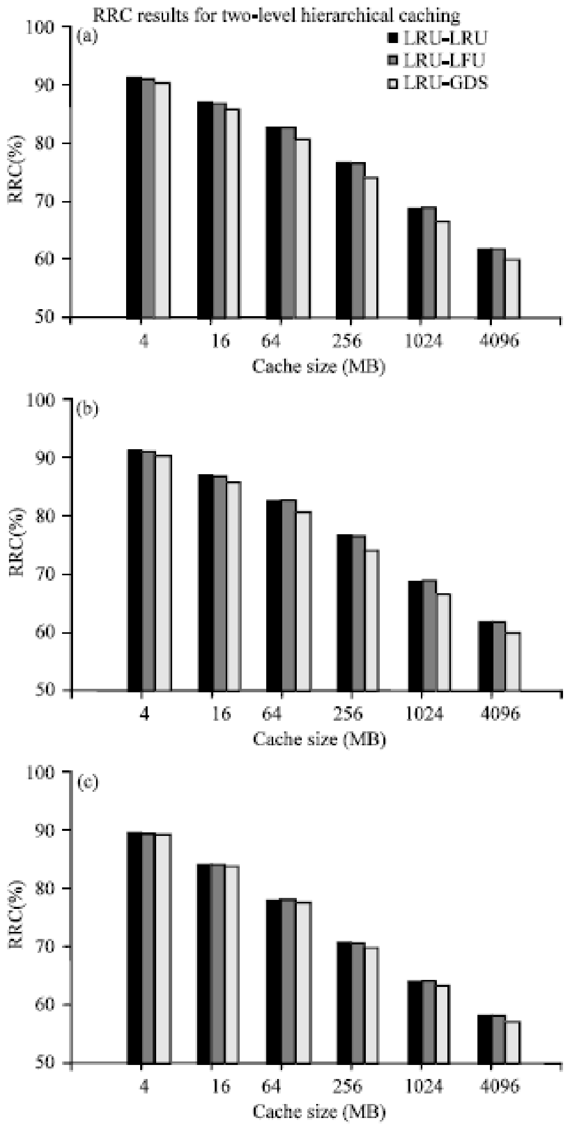
• If all the requests hit at the first-level cache, we can conclude RRC = RBC = 1/m. In two-level Web hierarchy caching model, we can conclude RRC = RBC = 1/3

• If all the requests are served at the server, then, RRC = RBC = 1. We can obtain RRC = RBC = 1 for two-level Web hierarchy caching model

In Web hierarchy caching model, the smaller the RRC or RBC is, the less the cost of requests is.

EXPERIMENT AND ANALYSIS

Web hierarchical caching model is an effective way to reduce the user perceived latency as well as resource consumption for network applications.



8.0 What is the bottom line; our recommendation:

The advantages of integrating caching and forwarding servers on an organization's network main stream to optimize network efficiency are significant:

- By using caching and forwarding servers, many organizations may find they do not need to upgrade their internet connections.
- Improved content delivery can add up to hours per month of time saved by knowledge workers.
- The internet is far more pleasant to use when response time are prompt.
- By using caching and forwarding servers, organizations are being good citizens and avoiding wasteful bandwidth usage.

- Internet congestion is going to get worse, not better, among the most optimistic forecasts concerning improvements in server hardware, router hardware and bandwidth; web caching is an essential part of any such strategy.

We strongly recommend that organizations adopt this practice as its benefits will provide enhanced performance and optimality.

Conclusion

A DNS system is a fundamental piece of the Internet framework. If DNS is unavailable, you'll have difficulty finding resources on the internet and, likewise, others will be unable to find you. That's because DNS is the phone book that translates names such as www.nytimes.com to internet protocol (IP) address such as 199.239.136.245, and vice versa. The hierarchical structure of the DNS name space, worldwide network of name servers, and efficient local caches allow broadband operators to provide high-speed, user-friendly Internet communications.

While the benefits of caching are more evident where a large number of users will potentially view the same content, studies indicate that even on LANs with as few as five users, performance gains can be impressive especially where the users tend to use the same information sources.

Caching and forwarding servers are critical components of a network infrastructure which guarantees fast speed network, bandwidth economy (reducing bandwidth costs), shorter time in content delivery (improving page load times) and increasing global availability of content.

References

[1] Behrouz, A. F., Sophia, C. F. 2007, Data Communication and Networking. Fourth Edition, Forouzan Networking Series. McGraw-Hill: New York
[2] Amadi E. C., Network Administration and Server Management: A Beginner's Guide. First Edition
[3] mk:@MSITStore:C:\WINNT\Help\DNSconcepts.chm::/sag_DNS_imp_UsingForward…10/2/2004
[4] http://www.vicomsoft.com/learning-center/lan-based-web-caching/
[5] https//technet.microsoft.com/en-us/library/cc754931.aspx
[6] http://www.appliedtrust.com/resources/infrastructure/understanding-dns-essential-knowledge-for-all-it-professionals
[7] https//technet.microsoft.com/en-us/library/cc730756.aspx
[8] mobile.severwatch.com/tutorials/article.ph/2193011/Back-To-Basics-DNS-Server-Roles—Cachingonly-Servers.html

