

On the Performance of Content Distribution Networks

Dakshi Agrawal, James Giles, and Dinesh C. Verma
T. J. Watson Research Center
International Business Machines Corporation
30 Saw Mill River Road
Hawthorne, NY 10532
{agrawal, jamesgiles, dverma}@us.ibm.com

Keywords: Content Distribution Network, Performance Analysis, User Response Time

ABSTRACT

This paper presents an analysis of the performance of content distribution networks. The effectiveness of a content distribution network depends on many factors, including the type of hosted application, cache-hit rate, average round-trip time between clients and servers, and the architecture of the content distribution network. Using a linear model for the client-perceived response time, we describe the effect of some of these factors and provide guidelines for evaluation of content distribution networks. In particular, we show that to reduce response time of a content distribution network compared to a single server system, a high cache-hit rate and careful placement of surrogates is required.

INTRODUCTION

Applications that are accessible over the World Wide Web usually follow a client-server paradigm, where the web-browser acts as the client program accessing a web-server to provide a set of pages. This set of pages may consist of unchanging information, such as static HTML pages, images, and audio-visual clips, or it may consist of dynamically generated information, such as pages generated as the output of a CGI script or a Java servlet. Quite often, the user-perceived response of the web-server is unacceptably large. In some cases the performance is poor due to an overload at the server. In other cases, the performance is poor due to the congestion within the Internet.

When poor performance is due to an overload at the server, the right approach to handle the performance problem is to increase the capacity of the server. However, an approach that can poor performance due to network congestion is still unclear. Among the possible solutions, various approaches have been suggested to manage the Quality of Service (QoS) within the network. The Integrated Services [3] approach provides a way for ap-

plications to reserve resources within the network by explicit signaling. The Differentiated Service [2] approach provides the notion of multiple types of classes within the network with some classes getting better treatment than others. However, due to the decentralized nature of the Internet, it is unclear when such QoS approaches will be deployed within the Internet, or even if they will be deployed at all.

An alternative approach to work around the performance problems in the Internet is to have a set of cooperating surrogates distributed within the Internet, with content from web-servers cached or replicated at the surrogates. This type of coordinated distribution scheme is known as a content distribution network. When a browser attempts to connect with a web-server, it is directed to one of the surrogates which is closer to its location. Such a redirection, known as request routing, can be implemented by means of an HTTP redirection [5] or by means of a modified Domain Name Server which provides the address of the appropriate surrogate instead of the origin server during the name resolution process [7]. Since the communication between the client and the surrogate is likely to be faster than between the client and the origin web-server, such an approach can be expected to result in better performance for the web-client. Several start-up companies have announced products and architectures which support this method of improving application performance. Examples include Akamai (with its Free Flow [1] architecture), SandPiper (with its Footprint architecture [9]), Digital Island, Edgix, iBeam, etc. A content distribution network is a system of such surrogates.

Despite recent growth in the number of content distribution networks and content distribution network providers, not much work has been done to understand the performance gains that can be achieved by this architecture. A comparison of the different request-routing schemes that can be used within CDNs is found in [4]. Qualitative claims about the gains of the content distri-

bution network architecture can be found in abundance at most companies that provide such a service. However, these claims usually appear to be marketing exercise rather than a serious analysis of the relative merits and demerits of content distribution networks.

In this paper, we attempt to make an analytical assessment of the effectiveness of content distribution networks. We will present a model for analyzing content distribution networks, and apply the model to different network scenarios. On the basis of the analytical model, we found several interesting characteristics about content distribution networks, and can identify the set of applications for which such networks would be advantageous, as well as identify those applications for which the content distribution networks are unlikely to result in any significant performance gains. To the best of our knowledge, this is the first attempt to understand the performance implications of content distribution networks in a systematic method.

The rest of this paper is organized as follows. We first describe factors not considered in this paper and provide two models for content distribution networks. Next, we present an analysis of response time and quantify the change in response time as different variables are changed. Finally, we summarize our results and describe guidelines drawn in this paper.

ASSUMPTIONS AND MODELS

The effectiveness of a content distribution network depends on many factors, including the type of hosted application, cache-hit rate, average round-trip time between clients and content distribution servers, and the architecture of the content distribution network. To analyze the effectiveness of content distribution networks, one needs to model all these factors, while keeping the mathematical analysis tractable. In this section, we will present a linear model for web-transactions and two models for content distribution networks. These models are based on the following simplifying assumptions:

Request Routing Request routing is the process of directing a client to a suitable server in a content distribution network. In this paper, we do not quantify the overhead incurred in different mechanisms of request routing.

Server Overload Server overload can dramatically change the response time of a web-transaction. Similarly, other malfunctions such as misconfiguration, or incorrect implementation of protocols can also increase response time. In this study, we assume that servers

in a content distribution network are lightly loaded and properly configured.

SYN Packet Loss In a congested network, TCP coarse-grained timeouts can be caused by the loss of connection setup packets, since the client has no way to know that setup packets have been lost. These coarse-grained timeouts, if frequent, can drastically increase client-perceived response time. We will assume that the number of connection request packets is small compared to the number of packets in the network, so that the impact of congestion is less significant.

Loss Probability We assume that loss rates change slowly over the life of a connection and the loss rates are independent of round-trip transmission times.

A Linear Model For Web-Transactions

In a related study, we conducted extensive experiments to model the client-perceived response time of typical web-transactions [6]. Our experimental testbed consisted of several personal computers connected by fast Ethernet switch. This testbed simulated a lightly loaded wide-area-network between a client and a server. In our experiments, neither the server nor the client was under heavy load. We considered the following web-transactions:

- Transfer of small (1K bytes), medium (100K bytes), and large (1M bytes) files.
- Transfer of HTTP pages with embedded objects.
- Servlets with different processing times.

Figure 1 illustrates experimental results for a basic servlet which returns a small file.

These experiments show that for a given application, the average client-perceived response time follows a linear model:

$$T = N\tau + P, \quad (1)$$

where τ is the average round-trip delay between the server and the client, and N and P are constants. The value of N is strongly correlated with the amount of data transferred from the server, while the value of P is correlated with the amount of processing done at the server to execute the web-transaction. Due to the strong correlation mentioned above, we will loosely refer to $N\tau$ as the *data transfer time* and refer to P as the *processing time*. Note that quantities in (1) are averages or expected values. A particular transaction may take longer or shorter time depending on different factors such as server load, state of software, etc.

This linear model can be intuitively explained by considering the steady-state bandwidth of TCP connec-

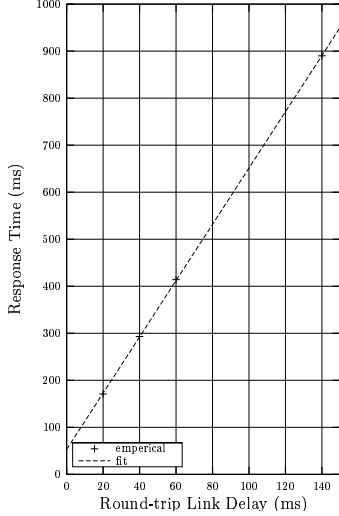


Figure 1: Response time versus round-trip delay

tions. It is well-known that the steady-state bandwidth B of a TCP connection is given by $B = C/\tau$, where C is a constant which depends on window size, packet drop probability, and other factors such as the particular version of TCP [8]. If coarse-grain timeouts are not frequent, then TCP connections will spend most time in steady state and the amount of time taken for a medium or a large size data-transfer would be proportional to τ . On the other hand, for small data-transfers (<10K bytes) which require only a few packets, most of the transfer time is spent in establishing the TCP-connection which again turns out to be proportional to the round-trip time τ for a given server load. Thus, (1) decomposes response time roughly into processing time and transfer time with transfer time being proportional to the round-trip delay. In the rest of this paper, we will use this linear model for the client-perceived response time.

Models of Content Distribution Networks

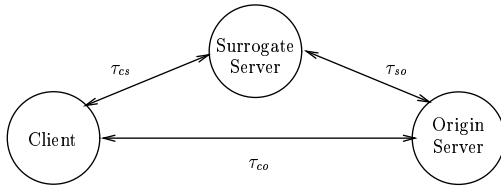


Figure 2: Round-trip times between servers and client

Consider a typical content distribution network with

an origin server and several surrogate servers. Assume that the average round-trip delay is τ_{co} between clients and the origin server, and the average round-trip delay is τ_{cs} between clients and their corresponding surrogate servers. Also assume that the average round-trip delay is τ_{so} between surrogates and the origin server (see Figure 2).

Without a content distribution network in place, clients will contact the origin server. Assuming the linear model of web-transactions holds, the average client-perceived response time is given by:

$$T_s = N_{co}\tau_{co} + P_{co} \quad (2)$$

where N_{co} and P_{co} are constant for a given application and server load when the client retrieves content directly from the origin server.

We consider two different architectures for content distribution networks based on how they handle requests that a surrogate is unable to satisfy.

Transparent Surrogate Architecture In this architecture, whenever a surrogate fails to satisfy a request, it contacts the origin server, gets appropriate data, and then serves the client request. Assuming that the surrogate fails to satisfy f fraction of the total requests, the average client-perceived response time is given by $T_p = (1-f)(N_{cs}^{hit}\tau_{cs} + P_{cs}^{hit}) + f((N_{so}\tau_{cs} + P_{so}) + (N_{cs}^{miss}\tau_{so} + P_{cs}^{miss}))$ where N_{cs}^{hit} and P_{cs}^{hit} describe the transaction between client and surrogate when the surrogate is able to satisfy the client request. Similarly, N_{so} and P_{so} describe the transaction between the origin server and the surrogate, and N_{cs}^{miss} and P_{cs}^{miss} describe the transaction between the surrogate and the client for requests that cannot be met by the surrogate alone. Note that the cache-hit rate is given by $100(1-f)$.

We will assume that the surrogate server needs approximately the same number of round-trips (or the same number of packet transmissions) to serve the client as the origin server, and so we assume that $N_{cs}^{hit} = N_{cs}^{miss} = N_{co}$. In most cases, if the surrogate cannot meet the client request, then it uses a succinct protocol to get the data from the origin server. The amount of data which a surrogate needs to transfer from the origin server will vary depending on the application. For example, it could be very small, characterized by $N_{so} = 3$, or 4 to fetch a user-profile from the origin server, or it could be large enough so that $N_{so} = N_{co}$. This, for example, would be the case when the surrogate simply fetches the required data from the server using the same mechanism as used by the client.

Translucent Surrogate Architecture In this model, a surrogate redirects clients to the origin server if it is unable to satisfy their requests. In this case, the average client-perceived response time is given by $T_l = (1 - f)(N_{cs}^{hit}\tau_{cs} + P_{cs}^{hit}) + f(N_{redir}\tau_{cs} + P_{redir}) + f(N_{co}\tau_{co} + P_{co})$ where N_{redir} and P_{redir} describe the redirection of client to the origin server.

We will again assume that a surrogate needs roughly the same number of round-trips (or the same number of packet transmissions) to serve the client as the origin server, and assume that $N_{cs}^{hit} = N_{co}$. Since the redirection can be done in two round-trip times (one round-trip to initiate the connection and one round-trip to send the request and get the redirection), we will assume that $N_{redir} = 2$.

We can examine the performance of a content distribution network by comparing the response time of a single server system to that of the content distribution network. For this comparison, we will use the ratio of the two response times $r_p = T_p/T_s$ and $r_l = T_l/T_s$ for the transparent surrogate architecture and translucent surrogate architecture respectively.

ANALYSIS OF RESPONSE TIME

The task of analyzing client-perceived response time is made complicated by the large number of variables present in the expressions for r_p and r_l . For easily interpretable results, it is necessary to eliminate or fix as many variables as possible and study the effect of changing the remaining variables. As a first step towards that goal, the following subsection shows that the processing time P_{cs}^{hit} , P_{so} and P_{redir} at the surrogate can be set equal to the processing time P_{co} at the origin server without much loss of insight into the behavior of content distribution networks.

Reducing Processing Time at a Surrogate

For some applications, it may be possible to reduce processing time at a surrogate (relative to the processing time at the origin server) by using clever software engineering, such as caching or by exploiting replication or subsetting. The extent of such reduction will depend on the application under consideration. In this subsection, we will examine the effect of reducing processing time at surrogates on the performance of a content distribution network. Our analysis shows that except in some narrow circumstances, reducing processing time at a surrogate relative to the processing time at the origin server is not likely to significantly reduce client-perceived response time.

We assume that the processing time at the surrogate is at least as small as the processing time at the origin server as in the case where the surrogate's infrastructure and equipment is the same as the origin server's. For the transparent surrogate architecture, in an optimistic case, we can assume that the processing time at a surrogate, given by P_{cs}^{hit} and P_{so} , can be made negligible compared to P_{co} . In this case, we can do a conservative analysis by assuming that $P_{cs}^{hit} = P_{so} = 0$. In a pessimistic case, we can assume that the processing time at the surrogate is as large as the processing time at the origin server, that is, $P_{cs}^{hit} = P_{so} = P_{co}$. In real life, the performance of an application design is likely to be closer to the pessimistic case. Under these assumptions on the processing times, the difference in r_p for these two extreme cases, $\Delta_1 = r_p^{optimistic} - r_p^{pessimistic}$, is given by

$$\Delta_1 = \frac{(1-f)P_{co} + fP_{co}}{N_{co}\tau_{co} + P_{co}} = \frac{1}{1 + \frac{N_{co}\tau_{co}}{P_{co}}}. \quad (3)$$

Similarly for the translucent surrogate architecture, for an optimistic case, we assume that $P_{cs}^{hit} = P_{redir} = 0$, and for a pessimistic case, we will assume that $P_{cs}^{hit} = P_{redir} = P_{co}$. Under these assumptions, the difference in r_l for the two extreme cases, $\Delta_2 = r_l^{optimistic} - r_l^{pessimistic}$, is given by

$$\Delta_2 = \frac{(1-f)P_{co} + fP_{co}}{N_{co}\tau_{co} + P_{co}} = \frac{1}{1 + \frac{N_{co}\tau_{co}}{P_{co}}}. \quad (4)$$

Thus, regardless of the architecture, the difference in the performance of the optimistic and the pessimistic case is given by $1/(1 + N_{co}\tau_{co}/P_{co})$. It turns out that the ratio of transfer time to processing time, $N_{co}\tau_{co}/P_{co}$, is an important factor in determining the performance of a content distribution network for an application. Henceforth, we will denote this ratio by Γ and refer to it as the *critical ratio*.

Clearly, the difference in optimistic and pessimistic case will be large if the transmission time $N_{co}\tau_{co}$ is much smaller than the processing time P_{co} . To get an idea of this difference, let us consider an application of (4).

Example 1 Assume that the difference Δ_1 or Δ_2 should be at least 0.15 before putting effort into improving the application-design at the surrogate. A simple calculation shows that this is feasible only for applications that have $N_{co}\tau_{co} \leq 5.7P_{co}$. For a typical round-trip delay of 60 ms (for a broadband client), and a processing time of 100 ms, we get $N_{co} < 10$. This value of N_{co} corresponds to very small transactions. For a narrowband client, τ_{co}

is likely to be larger and therefore the required value of N_{co} would be even smaller.

Thus, we conclude that applications with large processing time and small transmission time would benefit the most from being redesigned to achieve a smaller processing time.

In order to reduce the number of variables under consideration, we will make some simplifying assumptions. We will assume that $\tau_{co} = \tau_{cs} + \tau_{so}$. In practice, it is more likely that $\tau_{co} > \tau_{cs} + \tau_{so}$. However, the trends observed by making the above assumption will remain valid.

Since in real-life, the applications are more likely to be closer to the pessimistic case, for further analysis, we will assume that for the transparent surrogate architecture, $P_{cs}^{hit} = P_{so} = P_{co}$. Also let $P_{cs}^{miss} = \alpha_P P_{co}$, and $N_{cs}^{miss} = \alpha_N N_{co}$, and $\tau_{cs} = \alpha_\tau \tau_{co}$. Since $\tau_{so} = \tau_{co} - \tau_{cs}$, it follows that $\tau_{so} = (1 - \alpha_\tau) \tau_{co}$. In most circumstances, we can expect that $0 \leq \alpha_P, \alpha_N \leq 1.0$, while $0.1 \leq \alpha_\tau \leq 1.0$. $\alpha_\tau = 1.0$ corresponds to the surrogate being co-located with the origin server. On the other hand, we get $\alpha_\tau = 0.1$, if the origin server is located in another continent, while the surrogate is in the same community as the client.

Under these assumptions, we can derive the following expressions for r_p :

$$\begin{aligned} r_p &= \frac{(1-f)(N_{co}\alpha_\tau\tau_{co} + P_{co})}{N_{co}\tau_{co} + P_{co}} \\ &\quad + \frac{(f)(N_{co}\alpha_\tau\tau_{co} + P_{co}) + (f)(\alpha_N N_{co}\tau_{so} + \alpha_P P_{co})}{N_{co}\tau_{co} + P_{co}} \\ &= \frac{\Gamma(\alpha_\tau + f\alpha_N(1-\alpha_\tau)) + 1 + \alpha_P f}{\Gamma + 1} \end{aligned} \quad (5)$$

For the translucent surrogate architecture, we will assume that $P_{cs}^{hit} = P_{co}$. For an optimistic analysis in favor of content-distribution networks, we will also assume that the time taken for the redirection is negligible, and put $N_{redir} = 0$ and $P_{redir} = 0$. Under these assumptions, we have the following expression for r_l :

$$\begin{aligned} r_l &= \frac{(1-f)(N_{cs}^{hit}\tau_{cs} + P_{cs}^{hit}) + f(N_{co}\tau_{co} + P_{co})}{N_{co}\tau_{co} + P_{co}} \\ &= \frac{(f + (1-f)\alpha_\tau)\Gamma + 1}{\Gamma + 1} \end{aligned} \quad (6)$$

In the rest of this paper, we will use expressions

given by (5) and (6) for r_p and r_l respectively.

Effect of Surrogate Placement

It is intuitive to expect that putting more surrogates closer to the client population will decrease the response time experienced by the clients. However, multiple surrogates for a single hosted application also incur an overhead necessary to maintain coordination between them. In order to quantify this trade-off, we will evaluate the effect of reducing round-trip time between clients and their corresponding surrogates. We will show that $\alpha_\tau = \tau_{cs}/\tau_{co}$ is almost linearly related to the client-perceived response time.

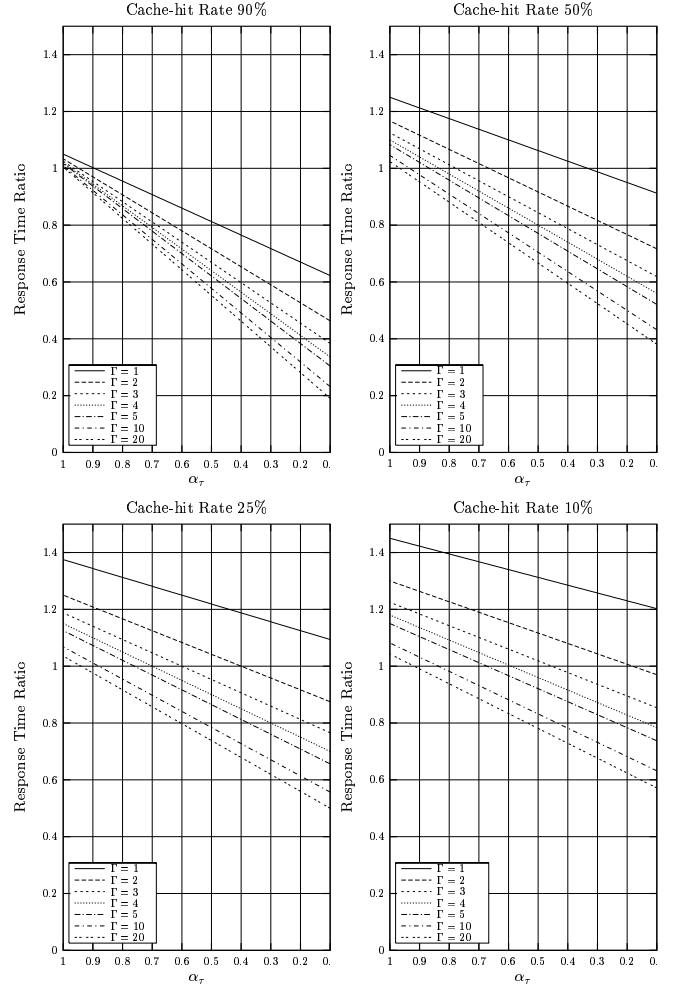


Figure 3: Almost linear decrease in r_p as α_τ decreases

For the transparent surrogate architecture, fix $\alpha_P = 1.0$, and $\alpha_N = 0.5$ so that the processing time for the client-surrogate message exchange with a cache miss is

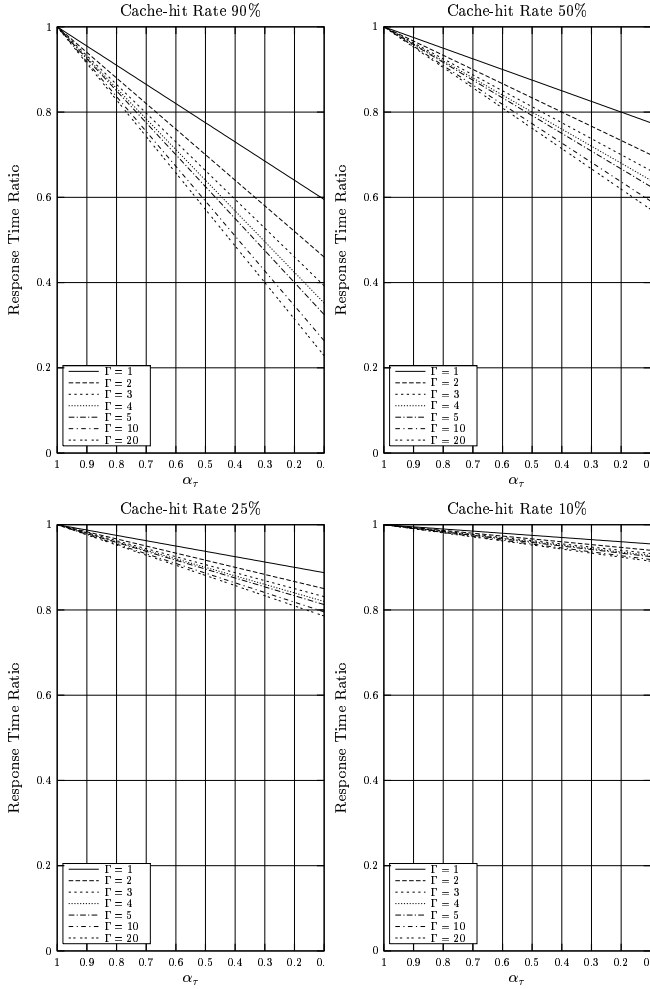


Figure 4: Linear decrease in r_l as α_τ decreases

the same as the processing time for the client-origin message exchange, but the number of messages is only half as many. It turns out that for other values of α_P and α_N , we obtain similar conclusions. Figure 3 and 4 show r_p and r_l respectively, as a function of α_τ for cache-hit rate of 90%, 50%, 25%, and 10%. Each figure has several plots corresponding to different values of the critical ratio $\Gamma = (N_{co}\tau_{co})/P_{co}$. In these figures the value of α_τ goes from 1 to 0.1. These figures show the following trends:

- Both r_p and r_l decrease almost linearly with α_τ . In fact, r_l decreases exactly linearly with α_τ .
- Performance of a content distribution network increases as the cache-hit rate increases. Later, the effect of cache-hit rate will be examined in more detail.

- A larger critical ratio Γ results in a more effective content distribution network. In other words, applications with relatively large amount of data-transmission and smaller processing time benefit the most by a content distribution network.
- In order to reduce the response time of a content distribution network significantly in comparison to the response time of a single server, it is essential to have large critical ratio Γ , small surrogate to client round-trip time, and large cache-hit rate.
- Even for 90% cache-hit rate, α_τ should be less than 0.5 to reduce response time of a content distribution network to half of that of a single server.

Effect of Critical Ratio

In order to study the effects of critical ratio, we assume that $\alpha_\tau = 0.5$, $\alpha_N = 0.5$, and $\alpha_P = 1.0$. We can fix the value of α_τ since its effect has already been established to be linear for a wide range of other parameter values. Figures 5 and 6 show the ratio of response times r_p and r_l respectively. Both ratios decrease monotonically as the critical ratio Γ increases. Hence applications that have a large critical ratio, that is, large transmission-time and small processing-time are more likely to benefit from a content distribution network. These figures also show that the performance of a content distribution network does not change much once the critical ratio of the application goes beyond 10.

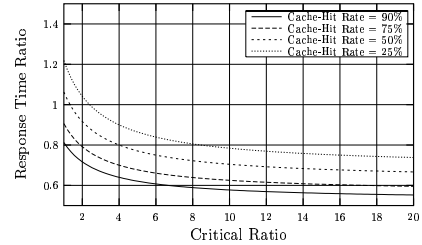


Figure 5: Critical ratio effect for transparent surrogate

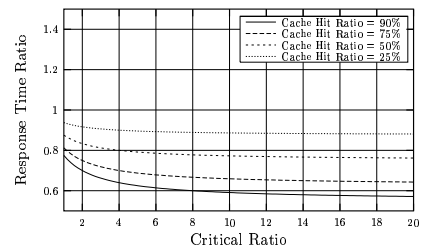


Figure 6: Critical ratio effect for translucent surrogate

Effect of Cache-Hit Rate

Figure 7 shows plots for response time ratio r_p as a function of cache-hit rate for $\alpha_\tau = 0.50$ and 0.33 . For each value of α_τ , we have several plots corresponding to critical ratio 1, 3, 5, and 10. This figure shows that the performance of a content distribution network increases as the cache-hit rate increases. By increasing cache-hit rate from 50% to 90%, we can decrease r_p by 0.15–0.45, depending on the critical ratio. Unfortunately, larger critical ratios result in less improvement by increasing cache-hit rates. For example, for $\Gamma = 10$, increasing the cache-hit ratio from 50% to 90% decreases r_p by 1.5 for $\alpha_\tau = 0.33$ and by 1.25 for $\alpha_\tau = 0.50$. Thus, having a large cache-hit rate is even more important for applications with smaller critical ratio.

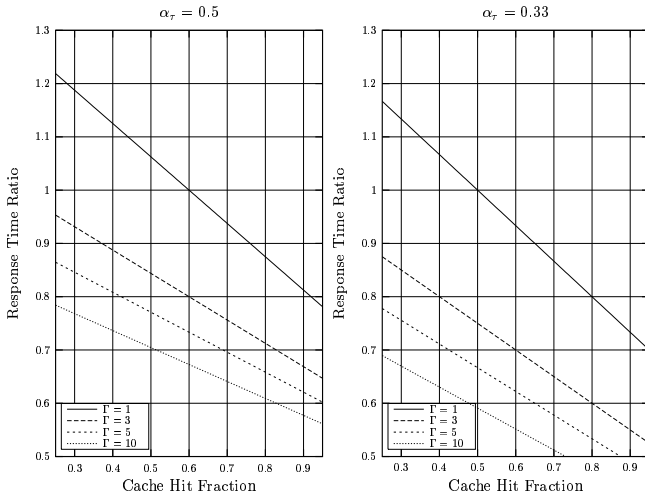


Figure 7: Cache-hit rate effect for transparent surrogate

Figure 8 shows similar plots for the translucent surrogate architecture. It shows that the decrease in r_l is more pronounced. For example, for $\Gamma = 10$, increasing cache-hit rate from 50% to 90% decreases r_l by almost 0.4 for $\alpha_\tau = 0.50$ and 0.33 . Thus, for the translucent surrogate architecture to be effective, we need high cache-hit rates.

Influence of CDN Architecture

The two architectures considered in this paper differ in the way that they deal with unsatisfied requests at surrogates. In the first architecture, the surrogate contacts the origin server and gets necessary data, while in the second architecture, the surrogate server redirects the client to the origin server. Arguably, the second architecture is simpler to implement, however the first architecture may still be preferred due to many reasons. For example, a surrogate may want to cache missing data for

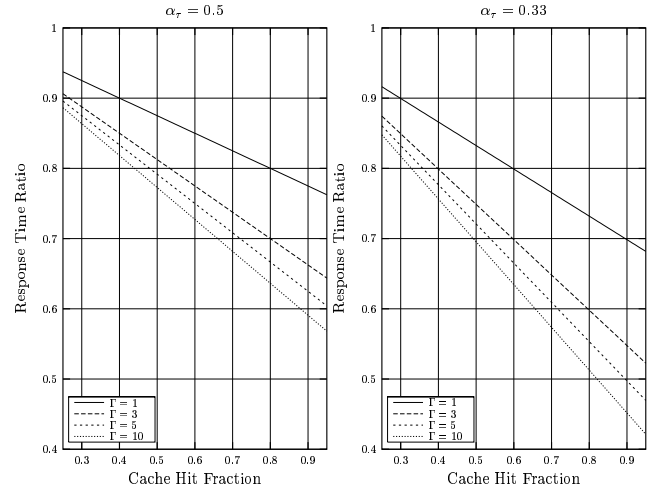


Figure 8: Cache-hit rate effect for translucent surrogate

future use, or it may only need to get a small amount of additional data to serve the client request. This may be the case when a surrogate does not have the personalization profile of a particular user, or when a small segment of the data has expired from its cache. While the first architecture is likely to incur more packet transfer than the transfers done by the second architecture for a given transaction, these packets are obtained over a round-trip time smaller than that of the second architecture. Thus we would expect that the transparent surrogate architecture will do better when α_τ and α_N are relatively small, and Γ is relatively large.

In order to compare the two CDN architectures, we will plot $r_p - r_l$ for some typical values of different parameters. Let us assume that $\alpha_\tau = 0.5$ and $\alpha_P = 0.5$. Figure 9 shows the difference, $r_p - r_l$, as a function of Γ for $\alpha_N = 0.5$ and $\alpha_N = 1.0$. For $\alpha_N = 0.5$, there is little difference between these two architectures since $|r_p - r_l| < 0.1$. The difference in performance between the two architectures becomes large as the cache-hit rate decreases. Also note that for small critical ratio Γ , the translucent surrogate architecture performs better than the transparent surrogate architecture. This is intuitively expected since a large value of Γ implies a large average round-trip time between the origin server and clients, which penalizes the translucent surrogate architecture. The case when $\alpha_N = 1.0$ is less favorable for the transparent surrogate architecture, since $\alpha_N = 1.0$ implies that surrogates get data from the origin using the same mechanism as the clients. Figure 9 show that the translucent surrogate architecture performs as well for this value of α_N , with the assumption that the redirection time is negligible.

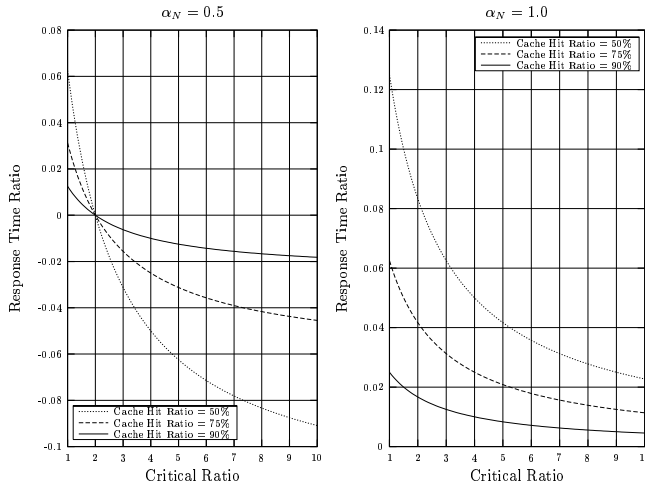


Figure 9: Comparison of architectures with $\alpha_\tau = 0.5$ and $\alpha_P = 0.5$

In all cases, we note that the difference between the two architectures is negligible if the cache-hit rate is as high as 75%. The choice becomes important with regard to response ratios only if the cache-hit rate is low.

Summary

The following is a summary of conclusions and guidelines drawn in this paper:

- Except in narrow circumstances of large processing time P_{co} and small transfer time $N_{co}\tau_{co}$, it may not be worth improving the processing time at the surrogates.
- In order to reduce the response time of a content distribution network significantly in comparison to the response time of a single server, it is essential to have a large critical ratio Γ , small average round-trip time between clients and surrogates, and a large cache-hit rate.
- The performance of a content distribution network increases linearly with the decrease in average round-trip time between clients and surrogates.
- A content distribution network is more effective for applications with high critical ratio Γ . However, the effectiveness is almost the same for applications with $\Gamma > 10$.
- High cache-hit ratio is more important for good performance of the translucent surrogate architecture than to good performance of the transparent surrogate architecture.
- The performance of a content distribution network can be increased by increasing the cache-hit rate.

However, the improvements decrease as the critical ratio increases. Hence, it is important to have high cache-hit rate for applications with low critical ratio.

- The choice of CDN architecture is likely to become important only if the cache-hit rate is less than 75%.
- A smaller α_N results in better performance for the transparent surrogate architecture. However, the improvement is limited to 0.1 for 75% cache-hit rate and to 0.2 for 50% cache hit rate.

Future work includes extending the model to include loss probability and classifying and identifying applications that would most benefit from various CDN architectures.

References

- [1] Akamai Technologies, Inc. *Freeflow content distribution service*. <http://www.akamai.com>.
- [2] Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z., and Weiss, W. December 1998. *An architecture for differentiated services*. RFC 2475. <http://www.faqs.org/rfcs/rfc2474.html>.
- [3] Braden, R., Clark, D., and Shenker, S. 1994. *Integrated Services in the Internet Architecture: an Overview*. RFC 1633. <http://www.faqs.org/rfcs/rfc1633.html>. (June).
- [4] Cain, B., Douglass, F., Green, M., Hofmann, H., Nair, R., Potter, D., and Spatscheck, O. 2000. *Known CDN request-routing mechanisms*. Internet draft: draft-cain-cdn-known-request-routing-00.txt. (Nov.).
- [5] Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and Berners-Lee, T. 1999. *Hypertext Transfer Protocol – HTTP/1.1*. RFC 2616. <http://www.w3.org/Protocols/rfc2616/rfc2616.html>. (June).
- [6] Giles, J., Agrawal, D., and Verma, D. C. *Modeling typical web transactions*. Under preparation.
- [7] InteliDNS. *DNS request-routing white paper*. <http://www.intelidns.com>.
- [8] Padhye, J., Firoiu, V., Towsley, D., and Kurose, J., 1998, “Modeling TCP throughput: A simple model and its empirical validation,” *Proceedings of ACM SIGCOMM* pp. 366–379.
- [9] Sandpiper Networks, Inc. *Footprint adaptive content distribution service*. <http://www.sandpiper.net> or <http://www.digitalisland.net>.