Caching and Optimized Request Routing in Cloud-based Content Delivery Systems[*]

Niklas Carlsson[a], Derek Eager[b], Ajay Gopinathan[1a], Zongpeng Li[c]

[a] *Linköping University, Sweden, niklas.carlsson@liu.se*
[b] *University of Saskatchewan, Canada, eager@cs.usask.ca*
[c] *University of Calgary, Canada, zongpeng@ucalgary.ca*

Abstract

Geographically distributed cloud platforms enable an attractive approach to large-scale content delivery. Storage at various sites can be dynamically acquired from (and released back to) the cloud provider so as to support content caching, according to the current demands for the content from the different geographic regions. When storage is sufficiently expensive that not all content should be cached at all sites, two issues must be addressed: how should requests for content be routed to the cloud provider sites, and what policy should be used for caching content using the elastic storage resources obtained from the cloud provider. Existing approaches are typically designed for non-elastic storage and little is known about the optimal policies when minimizing the delivery costs for distributed elastic storage.

In this paper, we propose an approach in which elastic storage resources are exploited using a simple dynamic caching policy, while request routing is updated periodically according to the solution of an optimization model. Use of pull-based dynamic caching, rather than push-based placement, provides robustness to unpredicted changes in request rates. We show that this robustness is provided at low cost – even with fixed request rates, use of the dynamic caching policy typically yields content delivery cost within 10% of that with the optimal static placement. We compare request routing according to our optimization model to simpler baseline routing policies, and find that the baseline policies can yield greatly increased delivery cost relative to optimized routing. Finally, we present a lower-cost approximate solution algorithm for our routing optimization problem that yields content delivery cost within 2.5% of the optimal solution.

Keywords:
Content delivery, Distributed clouds, Dynamic caching, Request routing optimization, Elastic storage

1. Introduction

A common approach to making content delivery applications scalable is to cache content at multiple geographically distributed sites, and to attempt to serve client requests from nearby servers. However, it is costly for a content provider to establish its own dedicated distributed infrastructure, and so content providers often rely on the services of CDN companies.

Recently another option has become possible, namely to make use of the facilities of cloud service providers. A number of major cloud service providers (e.g., Amazon[2], Google[3], Microsoft[4]), employ geographically distributed data centers that can be used to host content delivery servers. Unlike when using a CDN, the content provider can often retain control over content caching and request routing.[5] Unlike the

[1]Dr. Gopinathan worked on this project during his postdoc at Linköping University, but has since joined Google.
[2]http://aws.amazon.com/s3
[3]http://cloud.google.com
[4]http://www.windowsazure.com/
[5]In this paper, "request routing" refers to the choice of the site at which a request will be served, not to network-level routing.

case with dedicated infrastructure, use of cloud infrastructure enables server and storage resources to be dynamically acquired and released according to current demands.

When storage is sufficiently expensive that not all content should be cached at all server sites, policies are needed for determining what to cache where, and how to route client requests. A simple approach applicable to scenarios with non-elastic storage resources is to route to the server site nearest to the requesting client, and to use a pull-based caching approach together with a cache replacement policy such as LRU. Such a cache replacement policy assumes fixed storage resources rather than elastic resources. Also, this approach does not consider that it may sometimes be preferable to route client requests to some server site other than the nearest site, owing to a higher probability that the content is already cached at the other site. Another commonly-considered approach is to optimize semi-static content placement and request routing, according to known request rates (e.g., [1, 3, 23]). If actual request rates differ from those assumed in the optimization, however, the result could be highly sub-optimal.

In this paper, we consider distributed content delivery systems utilizing elastic storage resources. These resources are assumed to be obtained from a cloud service provider, with a cost incurred per GB per unit time. In practice, cloud service providers frequently advertise prices according to GB used per month, but base the charge for a month on the average usage over that month. We assume here that such averaging is done with a sufficiently fine time granularity that costs will accurately reflect the time-varying storage usage that results from dynamic caching.

We also assume that server and network resources can be dynamically acquired/released, and so, unlike much prior work, load balancing for the purpose of minimizing queueing delay is not an objective. Instead, our objective is to design caching and routing policies that minimize delivery cost as determined by the costs of acquired storage at each server site, the costs of loading caches with content, and the differential network delivery cost whenever client requests are routed to other than the nearest server site.

Our contributions are as follows:

- We propose a novel approach to content delivery using geographically distributed cloud platforms. Request routing is updated periodically through use of an optimization model, based on request rate estimates. Cache contents are updated dynamically, and are therefore able to adapt to unpredicted changes in request rates.
- The routing optimization problem we formulate is non-convex and standard techniques therefore are not directly applicable. However, we are able to identify and prove properties of the solution, leading to a feasible solution algorithm.
- We compare content delivery cost with our approach to that with optimal static placement and routing, as well as that with simple baseline routing policies that route all requests locally, or to the server site with the highest local request rate. Even with fixed request rates, use of dynamic caching is typically found to yield content delivery cost within 10% of that with the optimal static placement. We find that the simple baseline routing policies can yield greatly increased delivery cost, relative to the optimal routing policy with our proposed approach.
- We describe a lower-cost approximate solution algorithm for our routing optimization problem that yields content delivery cost within 2.5% of the optimal solution.

The remainder of the paper is organized as follows. Our system model is presented in Section 2. Section 3 describes our dynamic caching and optimized routing approach to content delivery. Section 4 characterizes the solution to our optimization model for request routing and presents a viable algorithm for finding it. Optimal static placement and routing is considered in Section 5. Section 6 presents numerical results and policy comparisons, while Section 7 considers lower-cost approximate solution algorithms for our routing optimization problem and evaluates their performance. Related work and conclusions are presented in Sections 8 and 9, respectively. Finally, to enhance readability, our major proofs are presented in appendices.

2. System Model

Since we assume elastic resources, cache storage and request routing can be considered separately for each file. Our model for caching and request routing for an individual file assumes a system with $N = |\mathcal{N}|$ server sites and $M = |\mathcal{M}|$ client locations[6], where $\mathcal{N}$ and $\mathcal{M}$ are the sets of server sites and client locations, respectively. We consider cache storage and routing at the level of sites, without concern for server/content replication within a site, and in the following refer to a "server site" simply as a "server". For simplicity, it is assumed that every client location $c \in \mathcal{M}$ is *local* to one server $i^*(c) \in \mathcal{N}$, and *remote* to all other servers $i \neq i^*(c), i \in \mathcal{N}$.

File delivery costs that are independent of cache storage and request routing need not be modelled. Our model includes three types of costs that are caching and/or routing dependent:

- *Remote routing cost* is the incremental cost associated with routing a client request for the file to a server other than the client's local server, *i.e.* to a remote server. It is modelled by a cost $R > 0$ that is independent of which remote server the request is directed to. Although this routing cost model incorporates only first-order location information (local vs. remote), previous work compared routing policy evaluations that also used such a first-order model to evaluations obtained by simulation, using transit-stub network topologies with differentiated link costs, and found good agreement [10].

- *Cache storage cost* is the cost of the storage resources used for the file. It is modelled by a cost $L > 0$ that is incurred per server, per unit of time that the file is in cache at that server.

- *Cache miss cost* is incurred when a client request is directed to a server at which the file is not currently cached, and the file must be retrieved from elsewhere (e.g., from a content provider server that stores the original copies of all content). Our model normalizes the incremental cost of a cache miss (versus cache hit) to be 1 cost unit.

The rate of requests from clients at location $c \in \mathcal{M}$ is denoted by λ_c. In our routing optimization model, we assume that requests from the M client locations can be modelled by independent Poisson processes, and that if the requests from a particular client location are split among multiple servers, request routing can be modelled as probabilistic. The rate of requests that are directed to server $i \in \mathcal{N}$ from client location $c \in \mathcal{M}$ is denoted by $\lambda_{c,i}$, the matrix of all $\lambda_{c,i}$ values is denoted by $\boldsymbol{\lambda}$, and $\gamma_i = \sum_{c \in \mathcal{M}} \lambda_{c,i}$ denotes the total rate of requests that are directed to server i.

3. Dynamic Caching and Optimized Routing

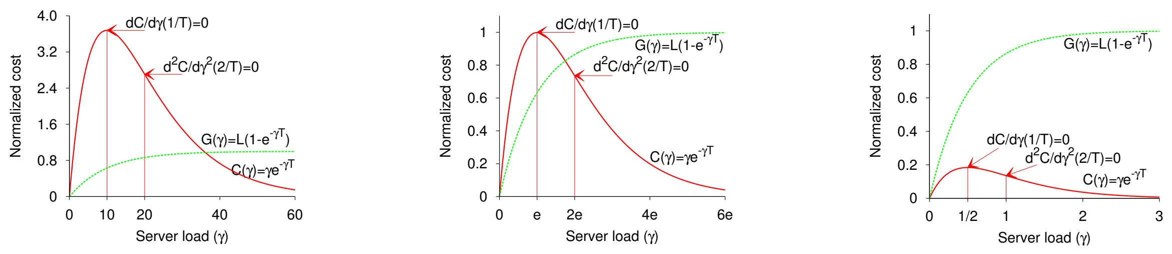
We propose an approach in which caching and routing policies operate at different time scales. Request routing is updated periodically using request rate estimates, through use of an optimization model. Cache contents are updated dynamically, and are therefore able to adapt to the unpredictable variations in request rates that can be expected in practice.

A pull-based approach to caching is used, wherein content is added to the cache at a server when accessed there, if not already present. A file is evicted when unaccessed at the server for a fixed period of time T, a policy parameter. With such a policy, cache storage use at a server tracks the file request rate there, increasing or decreasing correspondingly as the request rate increases or decreases. This policy also exploits any short-term temporal locality in the file request stream, as is often observed in practice.

Our optimization model for request routing incorporates (i) cache miss costs, (i) the costs of non-nearest server routing, and (iii) cache storage costs. Given our system model and dynamic caching policy, the probability that the file is not found in cache at server i when a request for the file is directed there is given by

$$P(\text{cache miss}) = e^{-\gamma_i T}, \tag{1}$$

[6]The "location" of a client might be defined according to what ISP the client obtains service from, for example, or may be defined at some other granularity.



(a) Miss cost dominates ($L = 1, T = 0.1$) (b) Equal peak costs ($L = 1, T = 1/e$) (c) Storage cost dominates ($L = 1, T = 2$)

Figure 1: Rates of incurring cache miss and storage costs at a server.

and therefore the rate $C(\gamma_i)$ at which server i incurs cache miss cost is

$$C(\gamma_i) = \gamma_i e^{-\gamma_i T}. \tag{2}$$

Figure 1 shows plots of the above function for varying total request rate at server i. It is straightforward to show that this function has a single inflection point at $\gamma_i = \frac{2}{T}$, and therefore is *strictly concave* for $\gamma_i < \frac{2}{T}$ and *strictly convex* for $\gamma_i > \frac{2}{T}$. The slope of this function is given by

$$\frac{dC(\gamma_i)}{d\gamma_i} = e^{-\gamma_i T}(1 - T\gamma_i). \tag{3}$$

Note also that there is a single maximum at $\gamma_i = \frac{1}{T}$. The above characteristics will be leveraged in our analysis characterizing optimal request routing.

The rate at which server i incurs cache storage cost is given by

$$G(\gamma_i) = L(1 - e^{-\gamma_i T}). \tag{4}$$

We note that this function is a monotonically increasing concave function.

Figure 1 shows three basic example scenarios in which (a) the cache miss cost dominates except under very high server load, (b) the peak rates of incurring cache miss cost and storage cost are equal, and (c) the cache storage cost dominates. Without loss of generality, we use $L = 1$ in Figure 1 as well as in all subsequent figures. With our choice of cost unit (equal to the cache miss cost), choosing $L = 1$ corresponds to taking as our unit of time the time it takes to incur a storage cost equal to the miss cost.

Finally, the rate at which server i serves requests from clients for which it is a remote server, multiplied by the incremental remote routing cost R for each such request, is given by

$$R \sum_{c \in \mathcal{M}: i^*(c) \neq i} \lambda_{c,i}. \tag{5}$$

Using expressions (2), (4), and (5) for the three cost components, optimal request routing can be obtained by finding $\lambda_{c,i}$ values that solve the following optimization problem:

Minimize

$$\textstyle\sum_{i \in \mathcal{N}} \left(\gamma_i e^{-\gamma_i T} + L(1 - e^{-\gamma_i T}) + R \sum_{c \in \mathcal{M}: i^*(c) \neq i} \lambda_{c,i} \right), \quad \text{where } \gamma_i = \sum_{c \in \mathcal{M}} \lambda_{c,i} \tag{6}$$

Subject to

$$\textstyle\sum_{i \in \mathcal{N}} \lambda_{c,i} = \lambda_c, \quad \forall c \in \mathcal{M}$$
$$\lambda_{c,i} \geq 0, \quad \forall i \in \mathcal{N}, \forall c \in \mathcal{M}$$

For convenience and without loss of generality, in the following we assume that $N = M$. The more general formulation can easily be transformed into this special case, by aggregating all of the client locations with the same local server into a single client location. We index servers and client locations from 1 to N such that client location i is local to server i, and such that $\lambda_i \leq \lambda_j$ whenever $i \leq j$. In the next section, we address the problem of how the above optimization problem is solved in order to obtain an optimal request routing.

4. Finding an Optimal Request Routing

Note that the optimization function (6) is neither convex, nor concave, and hence standard methods can not be applied. In Section 4.1 we consider some special cases for which the solution is readily obtainable. In Section 4.2 we derive some properties of the solution, under a mild constraint on the value of the policy parameter T, that enable general instances of the optimization problem to be solved in polynomial time.

4.1. Special Cases

We first consider the special cases in which the incremental remote routing cost R tends to zero or infinity:

- If $R \to 0$, then with an optimal request routing all requests are routed to a single server.
- If $R \to \infty$, then in the optimal request routing all requests are routed to the local server.

These observations are summarized in Theorem 1 and Theorem 2.

Theorem 1. *When the incremental remote routing cost $R \to 0$, then an optimal request routing will send all requests to a single server.*

Proof. (Theorem 1) We will prove this theorem by contradiction. Assume that $R \to 0$, and yet that with some optimal request routing $\boldsymbol{\lambda}^*$, the set of active servers, $\mathcal{N}' \subset \mathcal{N}$, is of cardinality at least two. Let i and j denote the indices of two such servers. Since, for positive total request rate, the cache storage cost function given in expression (4) is positive and strictly concave, while the cache miss cost function given in (2) is positive, strictly concave for total request rate less than $2/T$, and strictly decreasing for total request rate greater than $1/T$, both functions are strictly subadditive; i.e., for any positive γ_i and γ_j, $L(1-e^{-(\gamma_i+\gamma_j)}T) < L(1-e^{-\gamma_i T}) + L(1-e^{-\gamma_j T})$ and $(\gamma_i+\gamma_j)e^{-(\gamma_i+\gamma_j)} < \gamma_i e^{-\gamma_i T} + \gamma_j e^{-\gamma_j T}$. Therefore, a lower cost could be obtained by routing all of the requests being routed to these servers to only one of them, making the other inactive. This contradicts the assumed request routing optimality, and establishes the theorem. $\square$

Theorem 2. *When the routing cost $R \to \infty$, then in the optimal request routing each client request is served locally.*

Proof. (Theorem 2) The proof is trivial, as any request routing in which not all clients are served locally has an infinite cost, while the request routing in which all clients are served locally has a finite cost. $\square$

We next consider the special case in which the cost of a cache miss is negligible relative to the cache storage and remote routing costs L and R. In this case request routing concerns the tradeoff between reducing remote routing costs by selecting local servers, versus reducing cache storage costs by reducing the number of servers to which requests are routed. In this case our optimization problem reduces to:

$$\begin{aligned}
&\text{Minimize} \\
&\qquad \textstyle\sum_{i=1}^{N} \left(L(1-e^{-\gamma_i T}) + R(\gamma_i - \lambda_{i,i}) \right), \quad \text{where } \gamma_i = \textstyle\sum_{c=1}^{N} \lambda_{c,i} \qquad (7) \\
&\text{Subject to} \\
&\qquad \textstyle\sum_{i=1}^{N} \lambda_{j,i} = \lambda_j, \quad \forall j, 1 \leq j \leq N \\
&\qquad \lambda_{j,i} \geq 0, \quad \forall i,j, 1 \leq i \leq N, 1 \leq j \leq N
\end{aligned}$$

An optimal policy for this problem was first characterized for a somewhat different server selection system [10]. Below, we formalize this policy into a theorem in our context and provide a proof.

Theorem 3. *There is an optimal miss-oblivious request routing policy, in which for some index k ($1 \leq k \leq N$), all requests from client location i, for $k \leq i$, are served locally, while all requests from client location j, for $j < k$, are served at server N.*

Proof. (Theorem 3) Since the cache storage cost function given in expression (4) is nonnegative and strictly concave, it is optimal to serve all requests that receive remote service at a server with the highest rate of requests from its local clients (i.e., server N). Furthermore, it is optimal to serve either all requests from a client location at server N, or none, and the former case can hold only if all requests from client locations with equal or lower request rate are also served remotely. $\square$

To find an optimal request routing, one need only pick the lowest cost solution out of the N candidate solutions that satisfy the conditions of the above theorem. Hence, the total cost with optimal request routing is given by

$$\min_{k=1,2,\ldots,N}\left[L(1-e^{-(\lambda_N+\sum_{i=1}^{k-1}\lambda_i)T})+\sum_{i=k}^{N-1}L(1-e^{-\lambda_i T})+R\sum_{i=1}^{k-1}\lambda_i\right], \tag{8}$$

and an optimal request routing can be found in linear time.

4.2. General Case

We now consider the general case of the routing optimization problem under a mild constraint on the value of the policy parameter T. In particular, we impose the realistic constraint that T is chosen sufficiently large, such that the total of the costs at a server i, as given by the sum of expressions (2), (4), and (5), could never be *decreased* by increasing the arrival rate at server i of remotely-routed requests. By taking the derivative of this sum with respect to the arrival rate of remotely-routed requests, this constraint requires

$$R + e^{-\gamma_i T}(1 - T\gamma_i + LT) \geq 0 \quad \forall \gamma_i \geq 0. \tag{9}$$

Differentiating with respect to γ_i to find the value of γ_i that minimizes the left-hand side, yields the constraint

$$T \geq (\ln(1/R) - 2)/L. \tag{10}$$

Assuming the above constraint on T, we characterize and prove some properties of optimal request routing. This in turn allows us to clearly define a space of candidate solutions within which a solution to our optimization problem is guaranteed to exist, and to develop a polynomial-time algorithm for finding it.

We will say that a server i is *active* (for the file under consideration) if it has a non-zero rate of requests directed to it, *i.e.* $\gamma_i > 0$. Theorem 4 places an important constraint on the routing to an active server from its local clients.

Theorem 4. *With optimal request routing, it is never the case that only a fraction (less than one and greater than zero) of the requests from a client location are directed to the local server, with the remainder being routed elsewhere.*

A proof of this theorem is provided in Appendix A. Since it clearly could not be optimal for requests from remote clients to be routed to a server, while all requests from clients local to this server are directed elsewhere, Theorem 4 implies that with optimal request routing, each server must fall into one of three categories:

- *Local serving servers* are those that serve all requests for the file from local clients, but no other requests.
- *Inactive servers* do not serve any requests for the file. Requests from local clients are routed to remote servers.
- *Local and remote serving servers* are those that serve all requests for the file from local clients, as well as requests from remote clients.

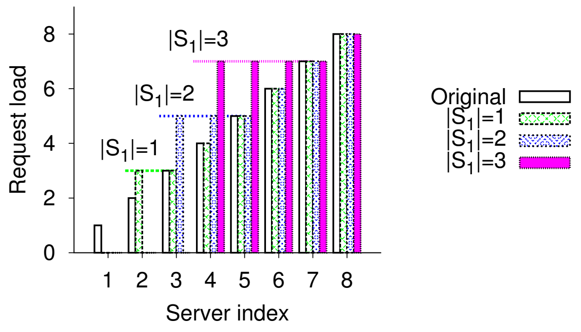


Figure 2: Server load-filling algorithm. ($N = 8$, $\lambda_i = i$, $|\mathcal{S}_2| = 0$)

Recall that we index servers and client locations such that client location i is local to server i, and such that $\lambda_i \leq \lambda_j$ whenever $i \leq j$. Theorem 5 states that with optimal request routing, servers can be divided into four distinct sets (one or more of which may be empty), such that for some i_1, i_2, and i_3, where $1 \leq i_1 \leq i_2 \leq i_3 \leq N+1$, servers with index $i < i_1$ are in set $\mathcal{S}_1$, index i such that $i_1 \leq i < i_2$ are in set $\mathcal{S}_2$, index i such that $i_2 \leq i < i_3$ are in set $\mathcal{S}_3$, and index i such that $i \geq i_3$ are in set $\mathcal{S}_4$. The servers in $\mathcal{S}_1$ are inactive. The servers in $\mathcal{S}_2$ and $\mathcal{S}_4$ are active, but serve only the requests from their local clients. In addition to the requests from their local clients, the servers in $\mathcal{S}_3$ also serve requests from remote clients (from the client locations local to the servers in $\mathcal{S}_1$). All servers in $\mathcal{S}_3$ have the same total rate of requests directed to them.

Theorem 5. *Optimal request routing can be achieved by a policy that partitions the servers into four sets, $\mathcal{S}_1$, $\mathcal{S}_2$, $\mathcal{S}_3$, and $\mathcal{S}_4$, one or more of which may be empty, such that:*

1. *For any $i \in \mathcal{S}_k$ and $j \in \mathcal{S}_l$ where $k < l$, we have $\lambda_i \leq \lambda_j$.*
2. *The servers in the sets $\mathcal{S}_2$ and $\mathcal{S}_4$ are local serving only. Servers in set $\mathcal{S}_1$ are inactive, while servers in $\mathcal{S}_3$ are local and remote serving.*
3. *All servers in $\mathcal{S}_3$ serve the same rate of requests, which does not exceed that of the least-loaded server in $\mathcal{S}_4$ (when $\mathcal{S}_4$ is nonempty).*

A proof of this theorem is provided in Appendix B. Note that a solution to our optimization problem can be found in polynomial time by considering each possible partitioning of the servers into the four sets $\mathcal{S}_1$, $\mathcal{S}_2$, $\mathcal{S}_3$, and $\mathcal{S}_4$, and then calculating the delivery cost with each partitioning. This approach would require $O(N^3)$ candidate solutions to be considered. Calculation of the delivery cost for each is $O(N)$, even when calculations for neighboring candidate solutions are leveraged.

The number of candidate solutions to be considered can be reduced by noting that once the set $\mathcal{S}_1$ is determined, so is the total rate of requests that will be served remotely. From Theorem 5, there is an optimal routing in which all servers in $\mathcal{S}_3$ serve the same rate of requests, which does not exceed that of the least-loaded server in $\mathcal{S}_4$. For a given total rate of requests to be served remotely, there is only one possible partitioning between the $\mathcal{S}_3$ and $\mathcal{S}_4$ servers that will satisfy these constraints, which can be found in linear time. Therefore, the number of candidate solutions that need be considered can be reduced to $O(N^2)$ (the number of possible choices of the sets $\mathcal{S}_1$ and $\mathcal{S}_2$), with a resulting total computational cost of $O(N^3)$. Figure 2 illustrates the allocation of request rate for differing numbers of servers in $\mathcal{S}_1$, with the remaining servers falling into either $\mathcal{S}_3$ or $\mathcal{S}_4$ (i.e., here $\mathcal{S}_2$ is assumed empty).

5. Optimal Static Placement and Routing

For comparison purposes, we now consider optimal static placement and routing. Under fixed request rates, and without any short-term temporal locality in the request streams (i.e., can be modelled by fixed-rate Poisson processes), intuitively it would be better to either permanently store the file at a server, or not store it at all at that server. In this section we first verify this intuition by proving that static placement is optimal for a generalization of our routing optimization problem in which each server i has its own T_i value; i.e., we prove that for each server i, the optimal T_i is either 0 or $\to \infty$. We then give an expression for the delivery cost with optimal static placement and routing. Finally, we observe that, while optimal under fixed-rate Poisson requests, static placement is not robust to changes in request rates.

Theorem 6. *Let K denote the number of client locations (possibly zero) with request rate λ_i at least $\frac{L}{\min[1,R]}$. Optimal caching and request routing can be achieved by a policy in which the K servers local to these client locations (those with index i such that $N-K < i \leq N$) serve the requests of their local clients with $T_i \to \infty$. When $K \geq 1$ and $R \leq 1$, the other servers (if any) are inactive, with their local clients served by one of the K servers. When $K \geq 1$ and $R > 1$, the other servers serve only their local clients with $T_i = 0$. When $K = 0$ and $L + \sum_{i=1}^{N-1} R\lambda_i \leq \sum_{i=1}^{N} \lambda_i$, all requests should be directed to the server with the highest local request rate (server N) with $T_N \to \infty$. Finally, when $K = 0$ and $L + R\sum_{i=1}^{N-1} \lambda_i > \sum_{i=1}^{N} \lambda_i$, all client requests should be served by the local server, with $T_i = 0$ for $1 \leq i \leq N$.*

Proof. (Theorem 6) This proof is derived on a case-by-case basis. Consider first an optimal value of T_i for server i. We will show that the objective function is minimized using either $T_i \to \infty$ or $T_i = 0$. To see this, consider the derivative of the objective function (6):

$$F(T_i) = \frac{d}{dT_i}[Expression(6)] = \gamma_i e^{-\gamma_i T_i}(L - \gamma_i). \tag{11}$$

This function is monotonically non-decreasing for $\gamma_i \leq L$ and monotonically non-increasing for $L \leq \gamma_i$. (Note that L, measured as cost per time unit, and γ_i, measured as requests per time unit, are related here by the normalized cost of a cache miss, equal to 1 cost unit.) This shows that any solution with intermediate T_i values is no better than one with either $T_i \to 0$ or $T_i \to \infty$. Hence, an optimal T_i value for server i is $T_i \to \infty$ or $T_i = 0$.

Second, assume that there exists at least one active server j with $T_j \to \infty$, and let us consider where the requests of the clients local to server i should be served. When $L \leq \min[1, R]\lambda_i$, the objective function (6) is minimized by serving all these requests locally with $T_i \to \infty$ (at cost L). To see this, note that the costs of operating with $T_i \to 0$ and $T_i \to \infty$ are γ_i and L, respectively. Using $T_i \to \infty$ and serving all these requests locally (at total cost L) is optimal in this case. (Note that any request directed to a different server would increase the cost by R.) When $\lambda_i < L$ and $1 \leq R$, the objective function (6) is minimized by serving all these requests locally with $T_i = 0$ (at cost λ_i). (Note that any request directed to a different server would increase the cost by $R - 1$.) Finally, when $\lambda_i R < L$ and $R < 1$, the objective function (6) is minimized by directing all these requests to server j. (Note that any request served locally would increase the cost by $1 - R$.)

At this point, we make two observations. First, from the above argument it is clear that in the optimal policy, the same decision should be made for all requests local to server i. If one request local to server i should be served locally, then all requests local to server i should be served locally, and if one such request should be served remotely, then all these requests should be served remotely. Second, as long as the request rate from at least one client location j is such that $L \leq \min[1, R]\lambda_j$, there will be at least one server (in particular, server j) that serves local requests with $T_j \to \infty$.

Based on the second observation, we must now consider the case where there is no server i with $L \leq \min[1, R]\lambda_i$. For this case, we first show that (i) there is no advantage to have more than one server with $T_i \to \infty$, and (ii) the server with the highest local request rate is the best candidate for such a server. The first property follows from the inequality $\sum_{i=N-K}^{N-1} \min[1, R]\lambda_i \leq (K-1)L$, which we use to show that:

$$L + \sum_{i=1}^{N-1} \min[1, R]\lambda_i \leq KL + \sum_{i=1}^{N-K} \min[1, R]\lambda_i. \tag{12}$$

The second property follows from the inequality $\lambda_c \leq \lambda_N$ $(1 \leq c \leq N)$, which we use to show that:

$$L + \sum_{i=1}^{N-1} \lambda_i R \leq L - \lambda_c R + \sum_{i=1}^{N} \lambda_i R. \tag{13}$$

In the case where there is no server i with $L \leq \min[1, R]\lambda_i$, the only two candidate solutions are that (i) all requests are served locally with $T_i = 0$ for all servers, or (ii) server N (with the highest local request rate) serves all requests and uses $T_N \to \infty$ (since for $T_N = 0$ it cannot be optimal for server N to serve remote requests). In the first case the total cost is $\sum_{i=1}^{N} \lambda_i$, and in the second case the total cost is $L + \sum_{i=1}^{N-1} \min[1, R]\lambda_i$. Enumerating each of the cases completes the proof. $\square$

Corollary 1. *With fixed-rate Poisson requests, the rate at which delivery cost is incurred with optimal static placement and routing is given by:*

$$\min\left[\sum_{i=1}^{N} \lambda_i, \min_{1\leq k\leq N}\left[kL + \min[1, R]\sum_{i=1}^{N-k} \lambda_i\right]\right]. \tag{14}$$

While static placement and routing is optimal with fixed-rate Poisson requests, we note that in many content delivery applications there is a high degree of popularity variation and churn [27, 6]. To illustrate the robustness to request rate changes that dynamic caching provides, in contrast to static placement, consider the extreme scenarios in which request rates are initially very low ($\lambda_i \to 0$) and then become very high (flash crowd), or vice-versa. For both very low and very high request rates, with both dynamic caching and optimal static placement, the optimal routing is to the local server.

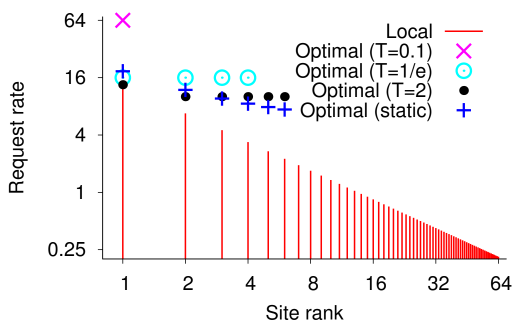
When request rates are very high, copies of the content should be placed at all servers, giving an optimal total rate at which delivery cost is incurred of NL. This is also the cost that would be incurred with dynamic caching. When request rates are very low, no copies should be cached, giving an optimal total rate at which delivery cost is incurred of $\sum_i \lambda_i \to 0$. With dynamic caching, this becomes approximately $\sum_i(\lambda_i + L(T/(T + 1/\lambda_i))) \to 0$.

In contrast, with static placement optimized for very low request rates (i.e. no copies), if request rates then became very high the total rate at which delivery cost would be incurred would be $\sum_i \lambda_i >> NL$. With static placement optimized for very high request rates (i.e. copies of the content placed at all servers), if request rates then became very low, the total rate at which delivery cost would be incurred would be $NL >> \sum_i(\lambda_i + L(T/(T + 1/\lambda_i)))$.

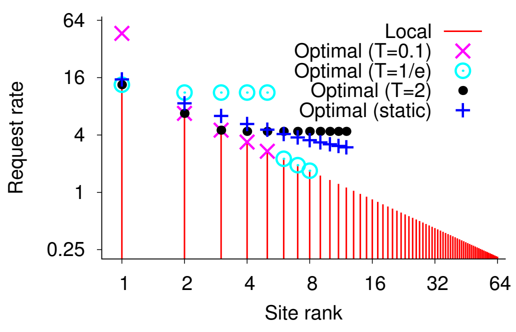
6. Numerical Results and Policy Comparisons

In this section we present numerical results, including delivery cost comparisons of dynamic caching and optimized routing to optimal static placement and routing, as well as to dynamic caching with simple baseline routing policies. We carry out these comparisons under different workload and cost conditions, assuming fixed-rate Poisson requests. Recall that the unit of cost is chosen as the incremental cost of a cache miss, and that choosing $L = 1$ corresponds to choosing as our time unit the time required to incur a storage cost equal to the miss cost.

Figure 3 shows the rate of requests directed to each server with dynamic caching and optimized routing for different values of the policy parameter T, as well as with local routing and with optimal static placement and routing. Results are shown for two different remote routing costs. The default value of $R = 0.5$ (Figure 3(a)) corresponds to a scenario where the cost of routing a request remotely is half the cost of a cache miss. The value of $R = 0.9$ (Figure 3(b)) corresponds to a scenario in which there is a high cost to routing remotely, not much less than the cost of a cache miss. The request rates from the client locations are distributed according to a Zipf distribution with parameter $\alpha = 1$, such that the average request rate across the client locations, denoted by λ, is 1. Request rate variations across the client locations could arise owing to differences in file popularity in different regions, or simply from differences in the number of clients at each location.



(a) Default ($R = 0.5$)



(b) High remote routing cost ($R = 0.9$)

Figure 3: Rates of requests directed to each server with different policies. (Default: $\lambda = 1$, $\alpha = 1$, $N = 64$, $L = 1$.)

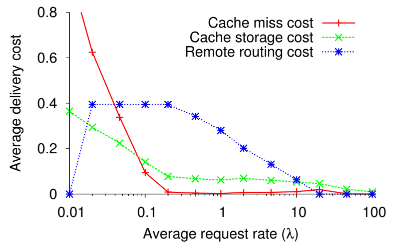
For a fixed request rate at a server, as T decreases the cache miss cost increases, creating incentive to concentrate requests at fewer servers. For $T = 0.1$, in fact, there is just a single active server in the optimal solution for the system considered in Figure 3(a). As T increases, there are increased numbers of active servers, but it is interesting to note that for $T = 2$ the number of active servers is the same in both Figures 3(a) and (b) as with $T \to \infty$, as seen by the results for optimal static placement and routing.[7] We have observed that $T = 1/e$ (using our normalized time unit, the point at which the peak rates of incurring cache miss cost and storage cost are equal, as illustrated in Figure 1), appears to achieve a good compromise between robustness to request rate changes, and cost performance under fixed-rate Poisson requests, and use it as the default in subsequent figures.

Finally, note that for most of the cases with dynamic caching and optimized routing considered in Figure 3, no more than three of the sets identified in Theorem 5 are non-empty. For example, with $T = 0.1$ set $\mathcal{S}_4$ is empty (as well as set $\mathcal{S}_2$ for the case of $R = 0.5$ considered in Figure 3(a)), and with $T = 2$ set $\mathcal{S}_2$ is empty. Among the considered cases, only with $T = 1/e$ and $R = 0.9$ are all four sets non-empty.

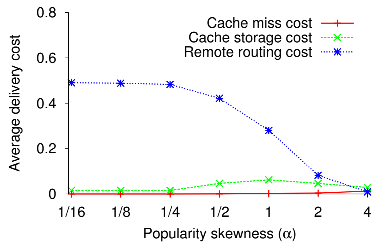
We next consider how the total delivery cost with dynamic caching and optimized routing is split among the three cost components. We report the cost incurred per processed request, i.e., the rate at which delivery cost is incurred, divided by the total request rate, and call this "average delivery cost". Figure 4 shows the average delivery cost for each cost component separately, for different workload, system, and policy parameter settings. Recall that the unit of cost is chosen as the incremental cost of a cache miss, and so, for example, a policy that always routed requests to the local server but never cached the file ($T = 0$) would have an average delivery cost of one. The average delivery cost could also substantially exceed one, for example in a scenario with very low request rates and a static placement policy that used $k > 0$ file copies incurring cache storage cost at rate kL. Results are shown varying (a) the average request rate λ per client location, (b) the request rate skew α, (c) the number of server sites N, (d) the value of the dynamic caching policy parameter T, and (e) the remote routing cost R. In all cases, we use the following default values: $\lambda = 1$, $\alpha = 1$, $N = 64$, $T = 1/e$, $R = 0.5$.

It is interesting to observe that frequently a substantial fraction of requests are routed to a remote server, as indicated by the relatively large remote routing costs. (Note that the fraction of requests that are remotely routed can be found by dividing the average remote routing cost by R.) Remote routing can concentrate requests at fewer servers, reducing the cache storage and cache miss costs. It is also interesting to observe that in Figure 4(d) all T values greater than 1 give the same results. This is consistent with the observation concerning Figure 3 that $T = 2$ gave the same number of active servers as $T \to \infty$. Finally, note that as the

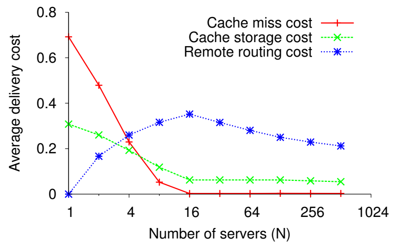
[7]Note that for optimal static placement and routing, it does not matter which of the servers with a copy of the file gets a remotely-routed request. For the results shown in Figure 3, remotely-routed requests have been spread evenly across all such servers.



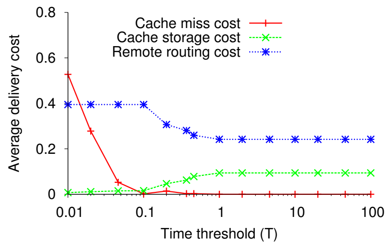
(a) Average request rate



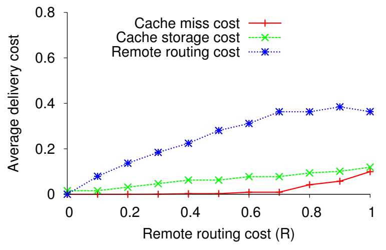
(b) Request rate skew



(c) Number of servers



(d) Policy parameter T



(e) Remote routing cost

Figure 4: Cost breakdown for dynamic caching and optimized routing. (Default: $\lambda = 1$, $\alpha = 1$, $N = 64$, $T = 1/e$, $R = 0.5$, $L = 1$.)

arrival rate increases in Figure 4(a), the average total delivery cost decreases. This is a consequence of our definition of average delivery cost as the rate at which delivery cost is incurred divided by the total request rate, and the ability to amortize cache miss and cache storage costs over multiple requests.

We now compare the average total delivery cost with dynamic caching and optimized routing, to that with optimal static placement and routing, and to that when using dynamic caching with two simple baseline routing policies:

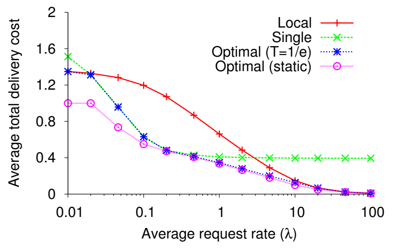
- *Local* routing in which all requests are directed to the local server.
- *Single* routing in which all requests are directed to the server with the highest local request rate.

Note that these baseline routing policies are optimal when $R \to \infty$ and $R \to 0$, respectively. Figure 5 shows these cost comparisons varying (a) the average request rate λ per client location, (b) the request rate skew α, (c) the number of server sites N, and (d) the remote routing cost R. The same default values are used as for Figure 4.

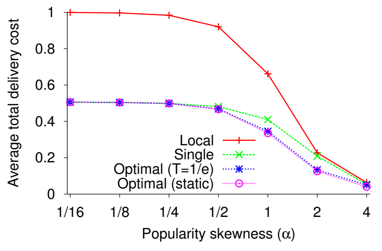
We note that optimized routing in many cases significantly outperforms the baseline policies. In fact, the performance difference can be shown to be unbounded. We also note that dynamic caching and optimized routing, with $T = 1/e$, typically yields delivery cost within 10% of that with the optimal static placement. Greater performance differences are only seen for small systems ($N < 8$) or low request rates ($\lambda < 0.1$ using our normalized time unit).

7. Lower-cost Routing Policies

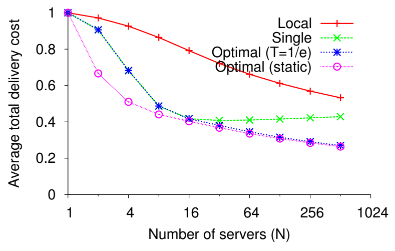
Although request routing based on a solution to the routing optimization problem formulated in Section 3 can substantially outperform simple baseline policies such as *local* and *single*, finding such a solution requires $O(N^2)$ candidate solutions to be considered. In this section we consider the cost penalties with two simpler policies, based on approximate solutions to the routing optimization problem:



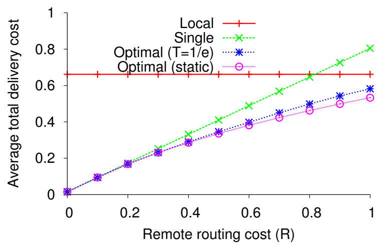
(a) Average request rate



(b) Request rate skew



(c) Number of servers



(d) Remote routing cost

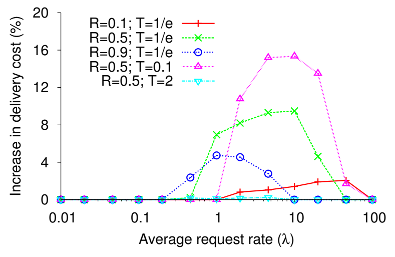
Figure 5: Cost comparisons. (Default: $\lambda = 1$, $\alpha = 1$, $N = 64$, $R = 0.5$, $L = 1$.)

- *Top-skewed* routing in which requests from clients local to the n servers with the lowest local request rates are directed to the server with the highest local request rate, which serves both its own local clients and all remote requests. All other servers serve only requests from local clients. The value of n is chosen that yields the smallest total delivery cost as given by optimization function (6).
- *Balanced* routing in which requests from clients local to the n servers with the lowest local request rates are directed to a set of servers with the next higher local request rates, so that all servers in this set serve the same rate of requests and none of them serve more than any other server. All other servers (with higher local request rate) serve only requests from local clients. Again, the value of n is chosen that yields the smallest total delivery cost as given by optimization function (6).

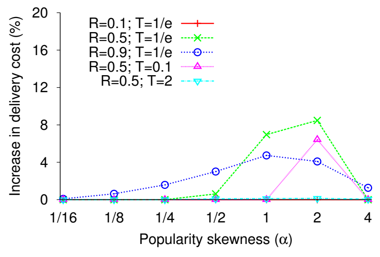
Note that the top-skewed policy has the same structure as the optimal miss-oblivious solution described in Section 4.1. The balanced policy corresponds to modifying our search algorithm in Section 4.2 by adding the constraint that $\mathcal{S}_2$ is empty (as was assumed when illustrating the algorithm operation in Figure 2). Note that with this modification only $O(N)$ candidate solutions need be considered, and the total computational cost is reduced to $O(N^2)$ from $O(N^3)$.

Figures 6 and 7 show the cost increase compared to the cost of the optimal solution for the top-skewed and balanced policy, respectively. The figures consider a wide range of scenarios, including (a) four orders of magnitude differences in average request rate λ, (b) balanced ($\alpha = 1/16$) to highly skewed ($\alpha = 4$) request rates, as well as (c) small ($N = 2$) to large ($N = 1024$) systems. Each figure shows results for three R values (0.1, 0.5, 0.9) and the default T value ($1/e$), and for three T values (0.1, $1/e$, 2) and the default R value (0.5).

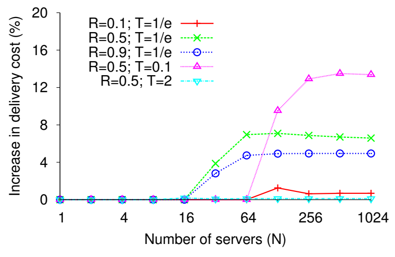
Figure 6 shows that there are cases where the cost increase for the top-skewed policy is up to 18%, although for $T = 1/e$ the cost increase does not exceed 10%. In contrast, the balanced policy consistently achieves a delivery cost within 2.5% of the optimal solution (Figure 7). The cost increase with this policy is greatest when the remote routing cost is high, as some of the servers that otherwise would be in server set $\mathcal{S}_2$ (and serve requests from local clients) instead are inactive. However, as the cost increase is small and the complexity of the policy is significantly less than that of the optimal solution, the balanced policy may be an attractive candidate to be used in practice.



(a) Average request rate

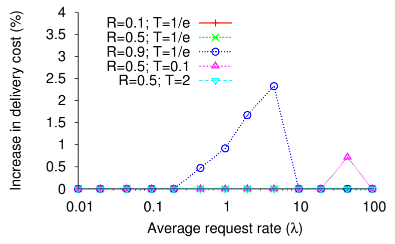


(b) Request rate skew

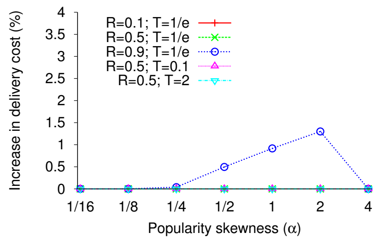


(c) Number of servers

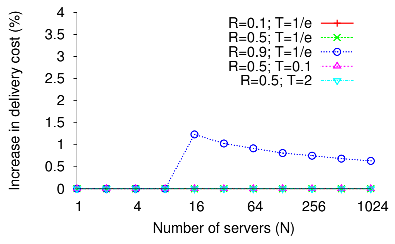
Figure 6: Cost increase if using the *top-skewed* policy compared to the optimal solution. (Default: $\lambda = 1$, $\alpha = 1$, $N = 64$, $L = 1$.)



(a) Average request rate



(b) Request rate skew



(c) Number of servers

Figure 7: Cost increase if using the *balanced* policy compared to the optimal solution. (Default: $\lambda = 1$, $\alpha = 1$, $N = 64$, $L = 1$.)

8. Related Work

Routing for geographically distributed cloud services is an area of increasing interest (e.g., [34]). The most closely related work, however, has focused on server (or replica) placement, and request routing, in more traditional content delivery platforms such as CDNs. Generally, replica placement can be divided into server placement and content placement. Both tasks consider the problem of deciding where to place service resources to satisfy a given demand.

In the context of server and replica placement, simple iterative greedy approaches that place one server or file at a time have proven to achieve close to minimal network costs [28, 24, 20, 29]. Other works have considered the problem of minimizing the overall network delay and its relationship to demand and server placement [8] or how to improve the system capacity by dynamically adjusting the number of replicas according to server load and demand [33]. There have also been a number of works that investigate how to efficiently distribute the content to the replica servers (e.g., [12, 25, 17]). In contrast to the above works, we assume that the candidate servers for a particular file are predetermined but that not all servers have cached copies, and consider the problem of optimized request routing among these servers.

Much work on content placement has considered hierarchical cache systems, evaluated with the aid of various mixed integer linear program (MILP) formulations, or variations thereof [3, 7, 14]. Borst et al. [7] analyze simple cooperative cache management algorithms designed to minimize bandwidth costs, while Applegate et al. [3] employ a Lagrangian relaxation-based decomposition technique combined with integer rounding to find near-optimal solutions. Huang et al. [19] extends these ideas to allow placement optimization on the granularity of segments, allowing them to better leverage impatient user video viewing patterns (where the users do not watch the full video) to reduce bandwidth usage. Other works (e.g., [23, 32]) have defined various optimization formulations, shown that the content placement problem is NP-complete, and provided heuristic replication strategies. In contrast to the above works, we consider the problem in which request routing is optimized and active replicas only exist while in the cache. For our specific problem, we leverage properties of the optimal solution to find ways to obtain the optimal solution efficiently.

There is a wide range of papers that consider the request routing problem, including works that consider both individual [11, 21, 35, 2, 30] and aggregated [1, 9, 15, 18, 10] service. With aggregated service multiple requests are more efficiently served as a group than individually. Similar to in our context, this scaling property results in tradeoffs between selecting the nearest server and directing requests to a single server that can achieve greater service aggregation. Within the context of streaming video delivery Almeida et al. [1] show that use of service aggregation can result in optimal solutions for placement, request routing, and routing of server multicasts that are very different from those for systems without service aggregation. Carlsson and Eager [9, 10] use a system model similar to that employed here, to compare classes of request routing policies for systems in which multiple servers implement a batched video-on-demand service, as well as for the case of energy/cost efficient content delivery using digital fountains.

Our paper uses a relatively general cost model, and we note that this model also is applicable to the context of energy-aware policies in systems with caches at the edge of the network. Here, the cache storage cost and cache miss cost correspond to the energy costs associated with keeping the content in the cache and loading the content into the cache, respectively. The remote routing cost also maps naturally, as edge networks typically consume much more energy than the core [5, 26, 4]. In this case, the remote routing cost reflects the difference between the energy consumed with local service (i.e., consuming a base energy cost staying within the same edge network) and with remote service (i.e., the cost associated with obtaining the content from a different edge network).

Among the works that assume individual service, perhaps the most closely related to ours are papers that consider optimized cache copy placement and request routing using TTL-based caches. TTL-based caches have been used and analyzed in the context of DNS servers [22]. However, in contrast to DNS servers, we assume that the content provider is in control of the content and therefore the "TTL" can be reset on each access as there is no risk of (uninformed) cache inconsistency. Other works have considered networks of LRU caches [13, 31] or TTL-based cache networks with exponentially distributed TTL values [16]. Here, we assume a fixed (deterministic) value of our parameter T.

9. Conclusions

In this paper we proposed an approach to content delivery using geographically distributed cloud platforms. In the proposed approach elastic storage resources are exploited using a simple dynamic caching policy, while request routing is updated periodically according to the solution of a cost optimization model. In comparison to static placement, dynamic caching provides greater flexibility in scenarios where there is a high degree of popularity variation and churn and/or short-term temporal locality in the request streams. Our optimization model for request routing considers (i) the dynamics of the content caching policy, (ii) the costs of non-nearest server routing, (iii) the costs of cache misses, and (iv) storage costs. We identified and proved properties of the optimal solution that allowed us to determine a viable algorithm for finding it, despite the general non-convexity of the problem.

Our numerical results suggest that, under fixed request rates, use of dynamic caching typically yields content delivery cost within 10% of that with the optimal static placement. We found that optimized routing can yield greatly reduced delivery cost compared to simple baseline routing policies. A lower-cost approximate solution algorithm for our routing optimization problem was found to yield close-to-optimal solutions.

10. Acknowledgements

This work was supported by funding from Center for Industrial Information Technology (CENIIT) at Linköping University and the Natural Sciences and Engineering Research Council (NSERC) of Canada.

References

[1] J. M. Almeida, D. L. Eager, M. K. Vernon, and S. J. Wright. Minimizing delivery cost in scalable streaming content distribution systems. *IEEE Transactions on Multimedia*, 6(2):356–365, Apr. 2004.

[2] M. Andrews, B. Shepherd, A. Srinivasan, P. Winkler, and F. Zane. Clustering and server selection using passive monitoring. In *Proc. IEEE INFOCOM*, New York, NY, June 2002.

[3] D. Applegate, A. Archer, V. Gopalakrishnan, S. Lee, and K. Ramakrishnan. Optimal content placement for a large-scale VoD system. In *Proc. ACM CoNext*, Philadelphia, PA, Nov. 2010.

[4] J. Baliga, R. W. A. Ayre, K. Hinton, and R. S. Tucker. Green cloud computing: Balancing energy in processing, storage, and transport. *Proceedings of the IEEE*, 99(1):149–167, Jan. 2011.

[5] R. Bolla, R. Bruschi, F. Davoli, and F. Cucchietti. Energy efficiency in the future Internet: A survey of existing approaches and trends in energy-aware fixed network infrastructures. *IEEE Communications Survey and Tutorials*, 13(2):223–244, 2nd Qtr. 2011.

[6] Y. Borghol, S. Mitra, S. Ardon, N. Carlsson, D. Eager, and A. Mahanti. Characterizing and modeling popularity of user-generated videos. In *Proc. IFIP PERFORMANCE*, Amsterdam, Netherlands, Oct. 2011.

[7] S. Borst, V. Gupta, and A. Walid. Distributed caching algorithms for content distribution networks. In *Proc. IEEE INFOCOM*, San Diego, CA, Mar. 2010.

[8] C. W. Cameron, S. H. Low, and D. X. Wei. High-density model for server allocation and placement. In *Proc. ACM SIGMETRICS*, Marina Del Rey, CA, June 2002.

[9] N. Carlsson and D. Eager. Content delivery using replicated digital fountains. In *Proc. IEEE/ACM MASCOTS*, Miami Beach, FL, Aug. 2010.

[10] N. Carlsson and D. L. Eager. Server selection in large-scale video-on-demand systems. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 6(1):1:1–1:26, Feb. 2010.

[11] R. L. Carter and M. E. Crovella. Server selection using dynamic path characterization in wide-area networks. In *Proc. IEEE INFOCOM*, Kobe, Japan, Apr. 1997.

[12] Y. Chawathe, S. McCanne, and E. Brewer. RMX: Reliable multicast in heterogeneous environments. In *Proc. IEEE INFOCOM*, Tel Aviv, Israel, Mar. 2000.

[13] H. Che, Y. Tung, and Z. Wang. Hierarchical web caching systems: Modeling, design and experimental results. *IEEE Journal on Selected Areas in Communications*, 20(7):1305–1314, Sept. 2002.

[14] J. Dai, Z. Hu, B. Li, J. Liu, and B. Li. Collaborative hierarchical caching with dynamic request routing for massive content distribution. In *Proc. IEEE INFOCOM*, Orlando, FL, Mar. 2012.

[15] Z. Fei, M. H. Ammar, and E. W. Zegura. Multicast server selection: Problems, complexity and solutions. *IEEE Journal on Selected Areas in Communications*, 20(7):1399–1413, Sept. 2002.

[16] N. C. Fofack, P. Nain, G. Neglia, and D. Towsely. Analysis of TTL-based Cache Networks. In *Proc. VALUETOOLS*, Cargese, France, Oct. 2012.

[17] S. Ganguly, A. Saxena, S. Bhatnagar, R. Izmailov, and S. Banerjee. Fast replication in content distribution overlays. In *Proc. IEEE INFOCOM*, Miami, FL, Mar. 2005.

[18] M. Guo, M. H. Ammar, and E. W. Zegura. Selecting among replicated batching video-on-demand servers. In *Proc. NOSSDAV*, Miami Beach, FL, May 2002.

[19] K.-W. Hwang, D. Applegate, A. Archer, V. Gopalakrishnan, S. Lee, V. Misra, K. K. Ramakrishnan, and D. Swayne. Leveraging video viewing patterns for optimal content placement. In *Proc. IFIP Networking*, Prague, Czech, May 2012.

[20] S. Jamin, C. Jin, A. Kurc, D. Raz, and Y. Shavitt. Constrained mirror placement on the Internet. In *Proc. IEEE INFOCOM*, Anchorage, AK, Apr. 2001.

[21] K. L. Johnson, J. F. Carr, M. S. Day, and F. Kaashoek. The measured performance of content distribution networks. *Computer Communications*, 24(2):202–206, Feb. 2001.

[22] J. Jung, A. W. Berger, and H. Balakrishnan. Modeling TTL-based Internet caches. In *Proc. IEEE INFOCOM*, San Francisco, CA, Apr. 2003.

[23] J. Kangasharju, J. Roberts, and K. W. Ross. Object replication strategies in content distribution networks. *Computer Communications*, 25(4):376–383, 2002.

[24] P. Krishnan, D. Raz, and Y. Shavitt. The cache location problem. *IEEE/ACM Transactions on Networking*, 8(5):568–582, Oct. 2000.

[25] G.-I. Kwon and J. W. Byers. ROMA: Reliable overlay multicast with loosely coupled TCP connections. In *Proc. IEEE INFOCOM*, pages 385–395, Hong Kong, China, Mar. 2004.

[26] P. Mahadevan, P. Sharma, S. Banerjee, and P. Ranganathan. A power benchmarking framework for network devices. In *Proc. IFIP Networking*, Aachen, Germany, May 2009.

[27] S. Mitra, M. Agrawal, A. Yadav, N. Carlsson, D. Eager, and A. Mahanti. Characterizing web-based video sharing workloads. *ACM Transactions on the Web*, 5(2):8:1–8:27, May 2011.

[28] L. Qiu, V. N. Padmanabhan, and G. M. Voelker. On the placement of Web server replicas. In *Proc. IEEE INFOCOM*, Anchorage, AK, Apr. 2001.

[29] P. Radoslavov, R. Govindan, and D. Estrin. Topology-informed Internet replica placement. *Computer Communications*, 25(4):384–392, Mar. 2002.

[30] S. Ratnasamy, M. Handley, R. Karp, and S. Shenker. Topologically-aware overlay construction and server selection. In *Proc. IEEE INFOCOM*, New York, NY, June 2002.

[31] E. J. Rosensweig, J. Kurose, and D. Towsley. Approximate models for general cache networks. In *Proc. IEEE INFOCOM*, San Diego, CA, Mar. 2010.

[32] X. Tang and J. Xu. On replica placement for QoS-aware content distribution. In *Proc. IEEE INFOCOM*, Hong Kong, China, Mar 2004.

[33] L. Wang, V. Pai, and L. Peterson. The effectiveness of request redirection on CDN robustness. In *Proc. OSDI*, Boston, MA, Dec. 2002.

[34] H. Xu and B. Li. Joint request mapping and response routing for geo-distributed cloud services. In *Proc. IEEE INFOCOM*, Turin, Italy, Apr. 2013.

[35] E. W. Zegura, M. H. Ammar, Z. Fei, and S. Bhattacharjee. Application-layer anycasting: a server selection architecture and use in a replicated Web service. *IEEE/ACM Transactions on Networking*, 8(4):455–466, Aug. 2000.

Appendix A. Proof of Theorem 4

Proof. (Theorem 4) The proof of Theorem 4 is by contradiction. Suppose to the contrary that with an optimal request routing, there is some active server i that serves only a fraction (less than one and greater than zero) of the requests from its local clients, with some non-zero fraction of the remainder of these requests being directed to a server $j \neq i$.

Consider the derivative of the total cost, as the rate at which requests from clients local to server i are directed to server j is increased, and the rate at which they are directed to server i is correspondingly decreased. With an optimal request routing, this derivative must equal zero; i.e., we must have

$$R + e^{-\gamma_j T}(1 - T\gamma_j + LT) - e^{-\gamma_i T}(1 - T\gamma_i + LT) = 0. \tag{A.1}$$

Relation (9) and the above equation imply that $1 - T\gamma_i + LT \geq 0$, and we also have (by assumption) $R > 0$. Together with the above equation, this implies that $\gamma_j > \gamma_i$.

We now show that total cost is higher with this request routing than it would be if all requests being routed to server i were routed to server j instead, thus contradicting the assumption that the request routing was optimal, and establishing the theorem. Note that the increase in the total remote routing cost when the requests being routed to server i are instead routed to server j is maximized when all of the requests being served by server i were local requests. Thus, we need only consider this case, i.e., we will show that

$$\gamma_i e^{-\gamma_i T} + L(1 - e^{-\gamma_i T}) + \gamma_j e^{-\gamma_j T} + L(1 - e^{-\gamma_j T}) - (\gamma_i + \gamma_j)e^{-(\gamma_i+\gamma_j)T} - L(1 - e^{-(\gamma_i+\gamma_j)T}) - R\gamma_i > 0. \tag{A.2}$$

We first show that the left-hand side of the above relation is minimized for $L = 0$, and thus we need only consider this case. Solving for R in equation (A.1) and substituting in relation (A.2), and then taking the derivative with respect to L yields

$$2 - e^{-\gamma_j T} - e^{-\gamma_i T} - (1 - e^{-(\gamma_i+\gamma_j)T}) - \gamma_i T e^{-\gamma_i T} + \gamma_i T e^{-\gamma_j T}. \tag{A.3}$$

Letting $s = \gamma_j/\gamma_i > 1$ and $t = \gamma_i T > 0$, this derivative can be re-written as

$$2 - e^{-st} - e^{-t} - (1 - e^{-t(1+s)}) - te^{-t} + te^{-st}. \tag{A.4}$$

To show that the left-hand side of relation (A.2) is minimized for $L = 0$, we need to show that expression (A.4) is greater than or equal to zero. Since the derivative of expression (A.4) with respect to s is negative for $t > 0$, we need only show that expression (A.4) is non-negative for $s \to \infty$. This follows immediately after recognizing that expression (A.4) for $s \to \infty$ is minimized for $t \to 0$.

For $L = 0$, after solving for R in equation (A.1) and substituting into relation (A.2), we get

$$\gamma_i e^{-\gamma_i T} + \gamma_j e^{-\gamma_j T} - (\gamma_i + \gamma_j)e^{-(\gamma_i+\gamma_j)T} - \gamma_i(e^{-\gamma_i T}(1 - T\gamma_i) - e^{-\gamma_j T}(1 - T\gamma_j)) > 0. \tag{A.5}$$

Letting $s = \gamma_j/\gamma_i > 1$ and $t = \gamma_i T > 0$ and simplifying yields

$$(s + 1)(1 - e^{-t}) - t(s - e^{t(s-1)}) > 0. \tag{A.6}$$

It is straightforward to show that the derivative of the left-hand side of the above relation with respect to s is positive for $s > 1$, $t > 0$. Therefore, we need only show that relation (A.6) holds when $s \to 1$ from above; i.e., that

$$2(1 - e^{-t}) > 0, \tag{A.7}$$

which follows immediately, completing the proof. □

Appendix B. Proof of Theorem 5

To prove Theorem 5, we first prove two properties that are required for the theorem to hold. These properties are given in Lemma 1 and Lemma 2. The first lemma concerns the unique property of the set $\mathcal{S}_3$. The second lemma concerns the boundary between set $\mathcal{S}_1$ and $\mathcal{S}_2$.

Appendix B.1. Balancing using remote requests

Lemma 1. *Let $\mathcal{S}$ be the set of servers that in addition to all of the requests from their local clients also serve requests from remote clients. There exists an optimal routing in which all servers in $\mathcal{S}$ serve the same total rate of requests; i.e., $\gamma_i = \gamma_j, \forall i, j \in \mathcal{S}$.*

To simplify the proof of this lemma we will first transform the optimization problem formulation using the substitution $y_i = \gamma_i - L$. Given this transformation (and the assumption that $M = N$, with servers and client locations indexed from 1 to N such that client location i is local to server i), we can now write the optimization problem as follows.

Minimize

$$LN + e^{-LT} \textstyle\sum_{i=1}^{N} \left(y_i e^{-y_i T} + e^{LT} R(L + y_i - \lambda_{i,i})\right), \quad \text{where } y_i = \textstyle\sum_{j=1}^{N} \lambda_{j,i} - L \tag{B.1}$$

Subject to

$$\textstyle\sum_{i=1}^{N} \lambda_{j,i} = \lambda_j, \quad \forall j, 1 \le j \le N$$
$$\lambda_{j,i} \ge 0, \quad \forall i, j, 1 \le i \le N, 1 \le j \le N$$

Note that if we ignore constants and set $R' = Re^{LT}$, the objective function can be written as

$$\sum_{i=1}^{N} \left(y_i e^{-y_i T} + R'(y_i - \lambda_{i,i})\right), \tag{B.2}$$

where we can identify the miss cost term $C(y_i) = y_i e^{-y_i T}$.

Proof. (Lemma 1) We will show that for any request routing in which two servers i and j in $\mathcal{S}$ serve different total request rates $\gamma_i \neq \gamma_j$, the total cost is no less than that of an alternative routing in which the requests from remote clients that are sent to these two servers are redistributed so that either i and j serve the same total rate of requests, or only one of them serves requests from remote clients. As this process can be repeated until all servers remaining in $\mathcal{S}$ serve the same total rate of requests, this will establish the lemma.

Let x denote the total rate of requests from remote clients that are directed to servers i and j, with a fraction f of these requests served by server i and the remaining fraction $(1-f)$ served by server j, for $0 \le f \le 1$. The total request rate for servers i and j is then given by $\gamma_i = \lambda_i + fx$ and $\gamma_j = \lambda_j + (1-f)x$, respectively, giving $y_i = -L + \lambda_i + fx$ and $y_j = -L + \lambda_j + (1-f)x$. The component of the total delivery cost that is dependent on f is given as

$$\begin{aligned} \mathcal{C} &= C(y_i) + C(y_j) \\ &= (\lambda_i + fx - L)e^{-(\lambda_i + fx - L)T} \\ &\qquad + [\lambda_j + (1-f)x - L]e^{-(\lambda_j + (1-f)x - L)T}. \end{aligned} \tag{B.3}$$

The derivative with respect to f of the above cost function is given as

$$\frac{d\mathcal{C}}{df} = x[e^{-y_i T} - T y_i e^{-y_i T}] - x[e^{-y_j T} - T y_j e^{-y_j T}], \tag{B.4}$$

while the second derivative is given as

$$\frac{d^2\mathcal{C}}{df^2} = x^2 T \left[(y_i T - 2)e^{-y_i T} + (y_j T - 2)e^{-y_j T}\right]. \tag{B.5}$$

The cost is minimized either when f is 0 or 1, or when expression (B.4) equals zero, in which case

$$e^{-y_iT} - Ty_ie^{-y_iT} = e^{-y_jT} - Ty_je^{-y_jT}, \tag{B.6}$$

implying that the slope of the cost function $C(y)$ at $y = y_i$ is equal to the slope at y_j. Therefore, the minimum cost can be achieved only when either: (i) both i and j serve the same total request rate, (ii) only one of i and j serve requests from remote clients, or (iii) neither (i) nor (ii) holds, but the slope of the cost function at y_i is the same as that at y_j. The result will be established if we can show that the minimum cost can never be achieved by (iii).

We distinguish two cases based on the sum of y_i and y_j (note that this sum is independent of f).

Case 1 ($y_i + y_j \leq \frac{4}{T}$): Assume without loss of generality that $y_i \leq y_j$. Using $y_j \leq \frac{4}{T} - y_i$, $y_i \leq y_j$ (by assumption), and $y_i \leq \frac{2}{T}$ (as implied by $y_i + y_j \leq \frac{4}{T}$ and $y_i \leq y_j$), together with expression (B.5) for the second derivative of the cost function (B.3) with respect to f, yields

$$\begin{aligned}
\frac{d^2\mathcal{C}}{df^2} &= x^2T\left[(y_iT-2)e^{-y_iT} + (y_jT-2)e^{-y_jT}\right] \\
&\leq x^2T\left[(y_iT-2)e^{-y_iT} + (2-y_iT)e^{-y_jT}\right] \\
&= x^2T(y_iT-2)(e^{-y_iT} - e^{-y_jT}) \\
&\leq 0,
\end{aligned}$$

with equality holding only when $y_i = y_j = \frac{2}{T}$. A point where the second derivative is less than zero could possibly achieve the minimum cost only if it was an endpoint ($f = 0$ or $f = 1$). So, the minimum cost cannot be achieved by (iii) in this case.

Case 2 ($\frac{4}{T} < y_i + y_j$): Consider the value of f (possibly outside the feasible interval [0, 1]) for which $y_i = y_j$. The first derivative of the cost function (B.3) is zero at this point since $y_i = y_j$, and the second derivative is positive since $y_i > \frac{2}{T}$ and $y_j > \frac{2}{T}$, implying that this value of f yields a local minimum of the cost function. Since there is only a single inflection point in the miss cost function $C(y)$, there can be at most one other value of f for which the slope of the miss cost function at y_i equals that at y_j. This cannot also be a local minimum of the cost function (B.3), however, since one cannot have two local minimums without an intervening maximum. Thus, the minimum cost cannot be achieved by (iii) in this case either, completing the proof. □

Appendix B.2. Inactive servers

Lemma 2. *There exists an optimal request routing in which a server j is never in the set of inactive servers when there is some server i that is local serving, and $\lambda_i \leq \lambda_j$.*

Proof. (Lemma 2) We prove this lemma by showing that with a routing in which server i is local serving and server j is inactive, with $\lambda_i \leq \lambda_j$, the delivery cost is not increased by one of the following two modifications to the routing: (i) the requests from the clients at location i are served remotely instead of locally; or (ii) the requests from the clients at location i are served remotely instead of locally and those from the clients at location j are served locally instead of remotely. Specifically, we determine a threshold request rate λ^*, such that whenever $\lambda_i \leq \lambda^*$, modification (i) does not increase the delivery cost, while when $\lambda^* < \lambda_i$, modification (ii) does not increase the delivery cost.

To calculate the change in total cost, the proof must take into account the load of the two servers, as well as the set of remote servers that serves i and/or j. Consider now the change in cost if the load of server i is moved remotely. First, the cost change δ_i at server i is

$$\begin{aligned}
\delta_i \equiv \delta(\lambda_i) &= \lambda_iR - \lambda_ie^{-\lambda_iT} - L(1-e^{-\lambda_iT}) \\
&\leq \lambda_iR - \lambda_ie^{-\lambda_iT}.
\end{aligned} \tag{B.7}$$

Note that the bound in the last line is tight for the special case when there is no cache storage cost (i.e., $L = 0$). As we will see, this corresponds to the most restrictive case.

In general, for $\lambda_i \leq -\frac{1}{T}\ln R$ this function is negative, so for these values it is clearly optimal for i to be served remotely. However, also for slightly larger values (particularly when $L > 0$) the increased cost may be offset by a reduction in cost at the remote servers that end up serving this client location. Second, with more servers providing additional flexibility, the change in cost due to the remote servers can be worst-case bounded by considering the scenario where these extra requests are directed to a single server r. This scenario will allow us to give a valid (but not necessarily tight) bound on the change in cost Δ_i at the remote server(s) from serving also the clients at location i. More specifically, assuming that the remote server r originally has a total rate of requests directed to it of γ_r, we have

$$\begin{aligned} \Delta_i \equiv \Delta(\lambda_i) &= (\gamma_r + \lambda_i)e^{-(\gamma_r+\lambda_i)T} - \gamma_r e^{-\gamma_r T} + L(e^{-\gamma_r T} - e^{-(\gamma_r+\lambda_i)T}) \\ &\geq (\gamma_r + \lambda_i)e^{-(\gamma_r+\lambda_i)T} - \gamma_r e^{-\gamma_r T}. \end{aligned} \tag{B.8}$$

Now, if we consider the total cost change $\delta_i + \Delta_i$, it is important to note that the change in the combined cache storage cost always is non-positive:

$$-L(1 - e^{-\lambda_i T}) + L(e^{-\gamma_r T} - e^{-(\gamma_r+\lambda_i)T}) = -L(1 - e^{-\gamma_r T})(1 - e^{-\lambda_i T}) \leq 0.$$

This shows that the special case when there is no cache storage cost (i.e., $L = 0$) is the most restrictive. To improve readability, for the remainder of this proof, we will therefore focus on this case. In this case, we do not have to keep track of L and the bound in relation (B.7) is tight.

Now, the bound λ^* for when it always is better to serve i remotely, can be calculated by solving the equation $\delta_i = -\Delta_i$. For the case when $2/T \leq \gamma_r$ this equation has a single solution. To see this, let us now look closer at the cost changes and their derivatives:

$$\begin{aligned} \frac{d\delta_i}{d\lambda_i} &= \frac{d}{d\lambda_i}[\lambda_i R - \lambda_i e^{-\lambda_i T}] \\ &= R - e^{-\lambda_i T}(1 - \lambda_i T), \end{aligned} \tag{B.9}$$

$$\begin{aligned} -\frac{d\Delta_i}{d\lambda_i} &= \frac{d}{d\lambda_i}[\gamma_r e^{-\gamma_r T} - (\gamma_r + \lambda_i)e^{-(\gamma_r+\lambda_i)T}] \\ &= e^{-(\gamma_r+\lambda_i)T}((\gamma_r + \lambda_i)T - 1). \end{aligned} \tag{B.10}$$

The first function is a monotonically increasing (positive) for $-\frac{1}{T}\ln R \leq \lambda_i$. The second function is monotonically decreasing (negative) when $2/T \leq \gamma_r$ and can be upper bounded by the cost derivative at the original point; i.e., $-\frac{d\Delta_r}{d\lambda_i} \leq e^{-\gamma_r T}(\gamma_r T - 1)$. These observations confirm that there only can be a single point above $-\frac{1}{T}\ln R \leq \lambda_i$ for which we have equality between the cost increase δ_i at server i and cost decrease $-\Delta_i$ at the remote server. As suggested above, we denote this point λ^*.

When $\lambda_i \leq \lambda^*$, the proof is trivial, as it is better for requests from client location i to be routed remotely. Consider instead the case when $\lambda^* < \lambda_i$. For this case, we want to show that it is better to change the solution such that i goes remotely and j is served locally. For this new solution to have a smaller cost, we must show that

$$C(\gamma_r) - C(\gamma_r + (\lambda_i - \lambda_j)) \geq \delta(\lambda_i) - \delta(\lambda_j). \tag{B.11}$$

This can be rewritten as

$$-\int_{\gamma_r}^{\gamma_r+(\lambda_i-\lambda_j)} \frac{dC}{dy}dy \geq \int_{\lambda_i}^{\lambda_j} \frac{d\delta}{d\lambda}d\lambda. \tag{B.12}$$

Now, to show this, it is sufficient to show that

$$\frac{dC}{dy}(\gamma_r - z) \leq \frac{d\delta}{d\lambda}(\lambda_i + z), \tag{B.13}$$

for all $0 \leq z$. We will show this using a two-step proof. We will first show that it is true for $z = 0$ and then use the fact that the derivative (as a function of z) of the two sides is strictly positive as well.

First, let us consider the case when $z = 0$. For this case we have that $\frac{dC}{d\gamma}(\gamma_r) = e^{-\gamma_r T}(\gamma_r T - 1)$. We next leverage the definition of the threshold rate λ^* to obtain a relationship between these variables and λ^*. More specifically, we use that $\delta(\lambda^*) = -\Delta(\lambda^*)$:

$$\begin{aligned}
\lambda^* R - \lambda^* e^{-\lambda^* T} &= \gamma_r e^{-\gamma_r T} - (\gamma_r + \lambda^*) e^{(\gamma_r + \lambda^*)T} \\
&= e^{-\gamma_r T}[\gamma_r - \gamma_r e^{-\lambda^* T} - \lambda^* e^{-\lambda^* T}] \\
&= e^{-\gamma_r T}(\gamma_r T - 1)(1 - e^{-\lambda^* T})\frac{1}{T} \\
&\quad + e^{-\gamma_r T}(1 - e^{-\lambda^* T} - \lambda^* T e^{-\lambda^* T})\frac{1}{T} \qquad \text{(B.14)}
\end{aligned}$$

From this expression we can now identify $\frac{dC}{d\gamma}(\gamma_r)$, and relate the two expressions as follows:

$$\begin{aligned}
\frac{dC}{d\lambda}(\gamma_r) &\equiv e^{-\gamma_r T}(\gamma_r T - 1) \\
&= \frac{\lambda^* T}{1 - e^{-\lambda^* T}}[R - e^{-\lambda^* T} + e^{-\gamma_r T} e^{-\lambda^* T}] - e^{-\gamma_r T} \\
&\leq R - e^{-\lambda^* T} + e^{-\gamma_r T} e^{-\lambda^* T} - e^{-\gamma_r T} \\
&\leq R - e^{-\lambda^* T}(1 - \lambda^* T) \\
&\equiv \frac{d\delta}{d\lambda}(\lambda^*). \qquad \text{(B.15)}
\end{aligned}$$

Finally, since $\frac{d\delta}{d\lambda}$ is monotonic, we have:

$$\frac{dC}{d\lambda}(\gamma_r) \leq \frac{d\delta}{d\lambda}(\lambda^*) \leq \frac{d\delta_A}{d\lambda_i}(\lambda_i) \qquad \text{(B.16)}$$

Having shown that relation (B.13) holds true for the case $z = 0$, we now want to show that it holds true for $0 \leq z$ as well. To do this we will show that the z-derivatives are monotonic (and the cost difference therefore will increase monotonically, as well). In other words, we want to show that

$$\frac{d}{dz}[\frac{dC}{d\lambda}(\gamma_r - z)] \leq \frac{d}{dz}[\frac{d\delta}{d\lambda}(\lambda_i + z)]. \qquad \text{(B.17)}$$

First, we expand the derivative into six terms:

$$\begin{aligned}
&\frac{d}{dz}[c - e^{-(\lambda_i + z)T}(1 - (\lambda_i + z)T) - e^{(\gamma_r - z)T}((\gamma_r - z)T - 1)] \\
&\qquad = e^{-(\lambda_i + z)T}(\frac{2}{T} + \lambda_i + z) + e^{-(\gamma_r - z)T}(\frac{2}{T} - \gamma_r + z). \qquad \text{(B.18)}
\end{aligned}$$

Note that there only is a single negative term; i.e., $\gamma_r e^{-(\gamma_r - z)T}$. On a case-by-case basis we next show that this term is smaller than some subset of the positive terms.

Case 1 ($\gamma_r \leq 2/T$): This case is trivial, as $2/T - \gamma_r \geq 0$, and we therefore have that both sets of terms are non-negative.

Case 2 ($2/T \leq \gamma_r \leq 4/T$): First, note that $\lambda_i + z \leq 1/T = 2/T - 1/T \leq \gamma_r - z$. Here, we have used that $z \leq 1/T - \lambda_i \leq 1/T$ and $2/T \leq \gamma_r$. Given these observations we can consider all positive terms without z:

$$\begin{aligned}
e^{-(\lambda_i + z)T}(\frac{2}{T} + \lambda_i) + e^{-(\gamma_r - z)T}\frac{2}{T} &\geq e^{-(\gamma_r - z)T}\frac{4}{T} \\
&\geq e^{-(\gamma_r - z)T}\gamma_r. \qquad \text{(B.19)}
\end{aligned}$$

The first inequality makes use of the above observations, and the last inequality makes use of the fact that $\gamma_r \leq 4/T$.

Case 3 ($4/T \leq \gamma_r$): For this case, we can use the fact that $4e^{-4} \geq \gamma_r e^{-\gamma_r T}$, to show that the first term always is greater than the negative term.

$$\begin{aligned} e^{-(\lambda_i+z)T} 2/T &\geq e^{-zT} 2/T \\ &= 4e^{-4} \frac{e^4}{2T} e^{-zT} \\ &\geq 4e^{-4} e^{-zT} 1/T \\ &\geq \gamma_r e^{-\gamma_r T} e^{-zT} \\ &= \gamma_r e^{-(\gamma_r - z)T}. \end{aligned} \tag{B.20}$$

This completes our proof. □

At this point we should note that the above lemmas imply that there exists an optimal solution in which $\gamma_i \leq \gamma_j$ for every pair of servers i and j for which $\lambda_i \leq \lambda_j$. This is formalized in Corollary 2. (For completeness, we include a separate proof of this Corollary.) Motivated by these observations we restrict ourselves to the optimal solution that preserves this ordering for the total loads; i.e., for which we also have $\gamma_i \leq \gamma_j$.

Corollary 2. *There exists an optimal solution in which $\gamma_i \leq \gamma_j$ for every pair of servers i and j for which $\lambda_i \leq \lambda_j$.*

Proof. (Corollary 2) Consider an optimal solution where this statement does not hold true.[8] In this case we must have (at least) two servers i and j for which $\gamma_i > \gamma_j$ and $\lambda_i \leq \lambda_j$. Now, consider an alternative solution in which the load of the two servers are exchanged such that server i serves γ_j and server j serves γ_i. We will show that this alternative solution has no greater cost and an optimal solution that satisfy the above conditions can be obtained by repeating this exchange process until there exists no such pair of servers.

First, it should be noted that the cache miss cost $C(\gamma_i) + C(\gamma_j)$ is independent of which server serves the most requests; i.e., $C_i(\gamma_i) + C_j(\gamma_j) = C_i(\gamma_j) + C_j(\gamma_i)$. Hence, there is no added cache miss cost associated with the alternative solution. Similarly, the cache storage cost is independent of which server serves the most requests; i.e., $G_i(\gamma_i) + G_j(\gamma_j) = G_i(\gamma_j) + G_j(\gamma_i)$.

Based on the above observations, any difference in cost would be due to differences in the amount of client requests being served locally. However, clearly there is no disadvantage to exchanging the loads of the two servers. To see this, consider the difference between the remotely served client requests before the switch $(\gamma_i - \min[\lambda_i, \gamma_i]) + (\gamma_j - \min[\lambda_j, \gamma_j])$ with that after the switch $(\gamma_j - \min[\lambda_i, \gamma_j]) + (\gamma_i - \min[\lambda_j, \gamma_i])$. To complete the proof we must hence show that:

$$\min[\lambda_i, \gamma_i] + \min[\lambda_j, \gamma_j] \leq \min[\lambda_i, \gamma_j] + \min[\lambda_j, \gamma_i]. \tag{B.21}$$

Now, consider the following three cases.

Case 1 ($\lambda_i \leq \gamma_i$ and $\lambda_j \leq \gamma_j$): In this case we have $\lambda_i \leq \lambda_j \leq \gamma_j$ and $\lambda_j \leq \gamma_j \gamma_i$. Hence, $LHS = \min[\lambda_i, \gamma_i] + \min[\lambda_j, \gamma_j] = \lambda_i + \lambda_j \leq \min[\lambda_i, \gamma_j] + \min[\lambda_j, \gamma_i] = RHS$.

Case 2 ($\lambda_i \leq \gamma_i$ and $\lambda_j > \gamma_j$): Due to our assumptions regarding active servers serving all local requests, we must hence have $\gamma_j = 0$. Furthermore, we note that both $\lambda_i \leq \lambda_j$ and $\lambda_i \leq \gamma_i$. Hence, $LHS = \min[\lambda_i, \gamma_i] + \min[\lambda_j, \gamma_j] = \lambda_i \leq \min[\lambda_j, \gamma_i] = \min[\lambda_i, \gamma_j] + \min[\lambda_j, \gamma_i] = RHS$.

Case 3 ($\lambda_i > \gamma_i$): Due to our assumptions regarding active servers serving all local requests, we must hence have $\gamma_i = 0$. Furthermore, as $\gamma_j < \gamma_i$, we must also have that $\gamma_j = 0$. Hence, $LHS = \min[\lambda_i, \gamma_i] + \min[\lambda_j, \gamma_j] = 0 = \min[\lambda_i, \gamma_j] + \min[\lambda_j, \gamma_i] = RHS$.

Clearly, serving these requests at server j rather than server i would be no worse, as none of the three costs would increase when exchanging the loads. This completes our proof. □

[8]Without loss of generality, we will allow the local load to be split across servers. (Clearly any improvements we can make to this more general case can be further improved according to Theorem 4.)

Appendix B.3. Putting the pieces together

Proof. (Theorem 5) First, we note that Lemma 2 ensures that only the servers with the lowest local request rates ($\mathcal{S}_1$) are inactive. The requests from clients local to these servers are directed remotely. Second, among the set of servers that are active, Lemma 1 (which states that the servers that are remote serving should have the same load) and Corollary 2 (which states that the loads in the optimal solution are monotonically non-decreasing) ensures that all servers in $\mathcal{S}_3$ are grouped together and they do not have a higher load than the least loaded server in $\mathcal{S}_4$. Third, since all servers in $\mathcal{S}_3$ have the same load, there can at most be two more sets of servers than those in sets $\mathcal{S}_1$ (which include the servers associated with the lowest local request rates) and $\mathcal{S}_3$ (which all have the same load). The set of servers in $\mathcal{S}_2$ and $\mathcal{S}_4$ are local serving (only), and simply consist of the remaining servers, which either are less loaded than those in $\mathcal{S}_3$ (these belong to $\mathcal{S}_2$) or are no less loaded than those in $\mathcal{S}_3$ (these belong to $\mathcal{S}_4$). Having identified and characterized the properties of each of the four groups, this completes the proof. $\square$