# Content Retrieval Delay Driven by Caching Policy and Source Selection

Mathias Björkqvist
IBM Research Zurich Laboratory,
8803 Rüschlikon, Switzerland
Email: mbj@zurich.ibm.com

Lydia Y. Chen
IBM Research Zurich Laboratory,
8803 Rüschlikon, Switzerland
Email: yic@zurich.ibm.com

***Abstract*—In this paper we study the content retrieval delay in a hybrid content distribution system, e.g., emerging content clouds [1], where a requested content item can be vertically retrieved from the central server and horizontally retrieved from network nodes. The content retrieval delay depends on the load intensities of the retrieval sources, which have asymmetric system properties such as bandwidth and cache capacity. The retrieval traffic arises due to heterogeneous content availability, i.e., content diffusion resulting from the applied caching policies, and the selection of retrieval sources. To optimize the retrieval delay, the advantages of the network nodes should be utilized while also leveraging the caching and retrieval capacity of the server. The traffic loads and latency of a given combination of source selection and caching policy is derived based on the content diffusion and distribution in the entire system. The simulation and analytical results show that satisfactory content retrieval delay is achieved when the retrieval selection is load-aware and the caching policies can effectively utilize the cache storage and retrieval capacity of both the network nodes and the server.**

## I. Introduction

Content distribution systems hosting wide varieties of content items have become ever popular in recent years. A hybrid architecture consisting of a central server and network/peer nodes has been shown to effectively deliver content items to end-users [3], [4]. Network and storage resources need to be deployed efficiently in order to provide satisfactory retrieval delay from the network nodes and the server, which have asymmetric retrieval and caching capabilities. Such a system design is very applicable and relevant to today's emerging content cloud systems, such as Amazon CloudFront [1]. Due to buffer limits for content caching, distributed network nodes can usually cache and serve a subset of the content items available in the system, whereas the central server caches all the content items. Caching policies manage the content availability in the network nodes, thereby also deciding the retrieval traffic. Moreover, the distribution of the retrieval traffic is also shaped by how a retrieval source is selected. As a result, the retrieval delay of a hybrid system hinges on the intensity of retrieval loads due to the caching policy as well as the source selection strategy.

Caching policies have been widely adopted in various computing systems to improve the content/data retrieval latency. Conventionally, a popularity-based caching policy is applied to optimize the hit ratio for a single node, so that its average latency of retrieving content items is minimized. Analytical caching studies [5] focus on scalability analyses of the caching capacity, especially in purely hierarchical systems. The performance of a caching strategy depends on the local content popularity, but also on the content distribution among the remaining network nodes. As a hybrid system consists of both hierarchical and P2P-like content retrievals, a well-designed caching policy needs to consider the popularity of content items, the limits on the caching capacity, as well as the retrieval capacities of the server and the network nodes.

Peer-to-peer like systems have been analytically proven to be highly scalable in terms of content provisioning when the number of peer/network nodes increases. A larger number of nodes generates more retrieval traffic, but simultaneously the peer retrieval capacity also grows. On the other hand, the complexity of the peer retrieval traffic and the management overhead also increases with the number of peers. Earlier studies on task assignment [6] have shown that a load-aware source selection strategy can benefit from highly distributed systems, whereas the performance of a system using load-oblivious source selection deteriorates when the number of peer nodes increases. As source selection and caching policies are clearly interrelated, both should be load-aware with regard to the peer network and server retrieval in order to minimize the retrieval delay.

In this study, we provide an analytical framework and develop a simulator to investigate the retrieval latency in a hybrid content distribution system. The retrieval delays are derived according to the source selection strategy and the caching policy. We consider two retrieval source selection strategies, Bernoulli selection and Shortest-Queue selection. The heterogeneous distribution of content items driven by caching policies is used for approximating the retrieval delay from multiple sources. Two caching policies are used as performance benchmarks for the proposed hybrid caching: a selfish policy, and an altruistic policy which adopts the proportional replication policy from [9]. The derivations of model and analysis are detailed in [2].

## II. System, Retrieval Selection, and Caching Strategies

The content distribution system considered consists of a central server and $N$ second tier nodes, referred to as the peer network, connected to the server and each other. In the remainder of this study, we use network nodes and peer nodes interchangeably when referring to the second tier nodes. Fig. 1 depicts the system schematics. The server has a buffer capacity that is sufficiently large to store all the available content items. The network nodes have limited storage capacity $B_p$, and they require a content management or caching policy to decide which content items to store and which to discard. The content management policies are further described later in this section. End users first send content retrieval requests to the network node to which they are connected. Each end user is connected to exactly one peer node. There is a total of $K$ unique content items, denoted by subscript $k = \{1 \dots K\}$. The total rate of requests received by a node from end users is $\lambda$, and the arrivals follow a Poisson distribution. All content requests are assumed to be uniformly distributed among the peer nodes. We also adopt the common observation [7] that the popularity of content item $k$, $r_k$, follows a Zipf distribution, $r_k = \frac{1}{k^\alpha}$. To facilitate further analysis, we normalize $r_k$, so that $\sum_k r_k = 1$. A single node thus receives requests for content item $k$ at the rate $\lambda r_k$, $\forall k$. Note that the popularity is assumed to be known by off-line profiling.

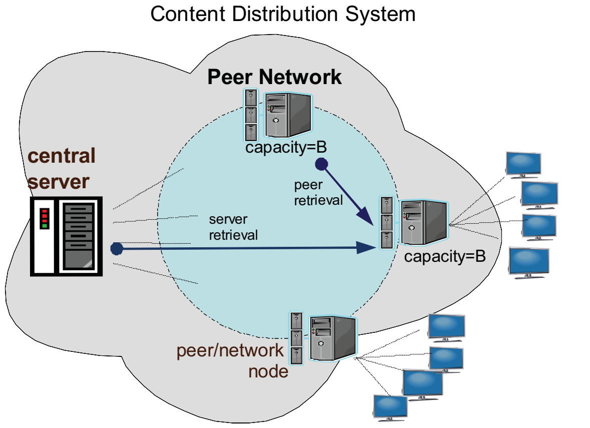


Fig. 1. System schematics of a hybrid content distribution system.

Upon receiving a request for a content item, a node satisfies the request from the local cache. If the content item is not available locally, the node queries the rest of the peer nodes in order to retrieve it. Among the network nodes with the requested content item, one is chosen according to the source selection criterions described in the following subsection. If a requested content item is not cached in any of the peer nodes, it will be retrieved from the server. Every network node can sustain $C_p$ horizontal peer retrieval connections, and the server has $C_s$ vertical connections. In this paper we assume that the server has a higher total bandwidth, and thus $C_s > C_p$.

The retrieval latency of a content item from a network node and the server is exponentially distributed with means $\mu_p =$ and $\mu_s$, respectively. When the number of retrieval requests exceeds the connection limits, i.e., $C_p$ or $C_s$, the request will join the respective waiting queue, from which requests are served following first-come, first-served scheduling. The content retrieval delay is therefore computed as the sum of the retrieval time and the queueing time in the retrieval channels. As it is out of scope for this study, we exclude any delay arising from the process of querying network nodes to find out if they are storing a desired content item from our calculations.

### A. *Caching Policies*

We summarize the optimal caching policies with respect to system and network architecture.

*1) Selfish Caching in a Hierarchical System:* In a conventional hierarchical content distribution system, content caching is thus designed to optimize for local requests only. We refer to this policy that optimizes the caching for local requests only as *selfish*. Each peer node statically stores the $B_p$ most popular content items with request rate $r_k, k = \{1 \dots B_p\}$. As every node keeps the same set of content items, there is no peer retrieval under such a policy. Requests for content items that are not cached locally will be satisfied by the server. It is straightforward to show that the static selfish policy generates the upper bound of the server load compared to other caching policies. We omit the formal proof due to page limit. One can expect the performance of a system using selfish caching to degrade with an increasing number of peer nodes. This is due to the increasing load on the server, whose retrieval capacity is constant.

*2) Altruistic Caching in a P2P-like System:* Tewari and Kleinrock [8], [9] showed that "proportional" replication, which keeps the number of content items in the system proportional to their request rates, can result in minimal content retrieval delays in a purely peer-to-peer system under certain assumptions. We refer to the proportional replication policy as *altruistic*. Here, the buffer space in all peer nodes is centrally controlled as a single, large buffer. All the peers collaborate tightly to maximize the overall peer network performance. Applying an altruistic policy in a hybrid system can provide the upper bound of the peer network size and load, as the server capacity may be underutilized. To implement the altruistic caching, we compute $n_k*$, the optimal number of copies of content item $k$ in the peer network, which in principle is proportional to the request rate $r_k$. As a node does not keep more than one copy of content item $k$, the maximum number of content item $k$ is thus $N$. The total number of content items cached in the system is equal to the total system capacity, $NB$. After determining the number of copies of each content item, we use the principle of load balancing to distribute those copies among all nodes.

*3) Caching in a Hybrid System:* Dynamic collaborative caching policies were proposed to minimize the retrieval traffic and bandwidth consumption of the peer network as well as the server [3], [4]. The collaborative caching policies acquire different degrees of information about the distribution of the content cached by peers. Consequently, the decision of whether or not to cache a content item needs to consider the trade-off between the local hit ratio, the peer hit ratio and the server hit ratio. However, the server and peer network retrieval capacities are usually not caching criteria considered in optimizing retrieval delay.

*4) Proposed Hybrid Caching:* We propose a load-aware hybrid caching policy, which partitions content items into 3 classes: (1) gold content; (2) silver content; and (3) bronze content, to each of which different caching strategies are applied. Thresholds, $T_1$ and $T_2$, are used to define each class. The gold content items, with $k \leq T_1$, are the ones with the highest request rates, and these content items are kept selfishly in all peer nodes. The bronze content items, where $k > T_2$, are the ones with the lowest request rates and they are not kept in the peer network at all. Content items where $T_1 < k \leq T_2$ are the silver class. These content items are always stored when received by a node. However, to make room for new content items, the silver items are also periodically discarded – the content item to discard is chosen either by a collaborative LRU (cLRU) policy, or randomly (Rnd). We name the two variants of the proposed hybrid caching policy *H-cLRU* and *H-Rnd*, respectively.

### B. Source Selection

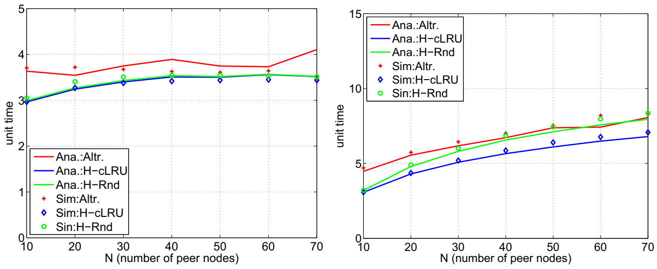
Source selection strategies have been well studied in the context of optimizing response time in web-server and P2P systems [3]. The selection can be load-oblivious or load-aware selection, the latter of which has been shown optimal in minimizing response time in generic multi-queue systems. In this paper, we use Bernoulli and Shortest-Queue selection among the network nodes, which are heterogeneous because of caching different content items.

1) Bernoulli Selection: From $n_k$ peers having content item $k$, one is selected with the probability of $\frac{1}{n_k}$. In static caching, the overhead of applying Bernoulli selection is negligible as the caching of $n_k \, \forall k$ items is fixed, whereas it has non-negligible overhead in dynamic caching because the content distribution changes.
2) Shortest-Queue Selection: From $n_k$ nodes having content item $k$, the node which has the lowest number of peer retrieval requests waiting in the queue is selected. The implementation overhead depends on finding $n_k$ in static and as well as dynamic caching. Therefore, it has the same order of implementation complexity as Bernoulli splitting in dynamic caching.

Note that existing analytical results regarding source selection are based on the assumption of homogenous peer content distribution. The analytical results of Shortest-Queue selections are based on approximation, especially for larger number of homogenous queues/peers. To analytically obtain the retrieval delay in the system considered here, the heterogeneous content distribution needs to factor into existing derivations of retrieval delay for both source selection strategies.

## III. Sample Results

A hybrid system serving $K = 300$ content items is considered. The central server has $C_s = 10$ retrieval connections with a mean retrieval time of $\frac{1}{\mu_s} = \frac{1}{10}$. Each peer node has $C_p = 1$ connection with a mean retrieval time of $\frac{1}{\mu_p} = \frac{1}{8}$. The request arrival rate from end-users at a node is $\lambda = 0.22$. The values of the system size $N$ are varied and caching capacity is $B = 20$. The average retrieval delays of Bernoulli selection and Shortest-queue selection obtained from the simulations and the derived analysis are depicted in Fig. 2.



(a) Bernoulli selection (b) Shortest-queue selection

Fig. 2. **Average retrieval delay**: analytical derivation vs. simulation results with $B = 20$.

## IV. Concluding Remarks

In this paper we investigate the scalability of retrieval delay in a hybrid system under different source selection strategies and caching policies. The steady-state analysis of content items derived from caching policies is used to compute the retrieval loads on the server and network nodes. The retrieval delay is then derived according to the loads and source selection among nodes with heterogeneous cached content distribution. The retrieval delay of the Shortest-Queue selection strategy is shown to strengthen the scalability of caching policies in large systems, whereas retrieval delay using Bernoulli selection decreases with increasing system size. For future work, we would like to further explore optimal threshold values of the hybrid caching policies and develop asymptotic results.

## References

[1] http://aws.amazon.com/cloudfront/.

[2] M. Björkqvist and Y. Chen. Content retrieval delay driven by caching policy and source selection. Technical report, IBM Research, 2010.

[3] S.C. Borst, V. Gupta, and A. Walid. Distributed Caching Algorithms for Content Distribution Networks. In *Proceedings of IEEE INFOCOM*, 2010.

[4] Y. Chen, M. Meo, and A. Scicchitano. Caching Video Content in IPTV Systems with Hierarchical Architecture. In *Proceedings of International Conference on Communications (ICC)*, 2009.

[5] A. Dan and D. Towsley. An Approximate Analysis of the LRU and FIFO Buffer Replacement Schemes. *SIGMETRICS Perform. Eval. Rev.*, 18(1):143–152, 1990.

[6] H-C Lin and C.S. Raghavendra. An Approximate Analysis of the Join the Shortest Queue (JSQ) Policy. *IEEE Trans. Parallel Distrib. Syst.*, 7(3):301–307, 1996.

[7] T. Qiu, Z. Ge, S. Lee, J. Wang, Q. Zhao, and J. Xu. Modeling Channel Popularity Dynamics in a Large IPTV System. In *Proceedings of SIGMETRICS*, pages 275–286, 2009.

[8] S. Tewari and L. Kleinrock. On Fairness, Optimal Download Performance and Proportional Replication in Peer-to-Peer Networks. In *IFIP Networking*, pages 709–717, 2005.

[9] S. Tewari and L. Kleinrock. Proportional Replication in Peer-to-Peer Networks. In *Proceedings of IEEE INFOCOM*, 2006.