International Journal of Research Studies in Science, Engineering and Technology
Volume 3, Issue 1, January 2016, PP 1-5
ISSN 2349-4751 (Print) & ISSN 2349-476X (Online)

Content Caching and Scheduling For Clusters with Elastic and Inelastic Traffic

[1]Syed Allahbaksh, [2]Syed.Akhtar Basha, [3]Sd Abdul Haq

[1]PG Scholar, Department of CSE, QCET, Nellore
[2]Associate Professor, Department of CSE, QCET, Nellore
[3]Associate Professor, Department of CSE, QCET, Nellore

Abstract: *The rapid growth of wireless content access implies the need for content placement and scheduling at wireless base stations. We study a system under which users are divided into clusters based on their channel conditions, and their requests are represented by different queues at logical front ends. Requests might be elastic (implying no hard delay constraint) or inelastic (requiring that a delay target be met). Correspondingly, we have request queues that indicate the number of elastic requests, and deficit queues that indicate the deficit in inelastic service. Caches are of finite size and can be refreshed periodically from a media vault. We consider two cost models that correspond to inelastic requests for streaming stored content and real-time streaming of events, respectively. We design provably optimal policies that stabilize the request queues (hence ensuring finite delays) and reduce average deficit to zero [hence ensuring that the quality-of-service (QoS) target is met] at small cost. We illustrate our approach through simulations.*

1. INTRODUCTION

The resource abundance (redundancy) in many large data centers is increasingly engineered to offer the spare capacity as a service like electricity, water, and gas. For example, public wireless network service providers like Amazon Web Services virtualizes resources, such as processors, storage and network devices, and offer them as services on demand, i.e., infrastructure as a service (IaaS) which is the main focus of this paper. A virtual machine (VM) is a typical instance of IaaS. Although a VM acts as an isolated computing platform which is capable of running multiple applications, it is assumed in this study to be solely dedicated to a single application, and thus, we use the expressions VM and application interchangeably hereafter. Wireless network services as virtualized entities are essentially elastic making an illusion of "unlimited" resource capacity. This elasticity with utility computing (i.e., pay-as-you-go pricing) inherently brings cost effectiveness that is the primary driving force behind the wireless network. In this project, address the issue of disk I/O performance in the context of caching in the wireless network and present a cache as a service (CaaS) model as an additional service to IaaS. For example, a user is able to simply specify more cache memory as an additional requirement to an IaaS instance with the minimum computational capacity (e.g.,micro/small instance in Amazon EC2) instead of an instance with large amount of memory (high-memory instance in Amazon EC2). The key contribution in this work is that our cache service model much augments cost efficiency and elasticity of the wireless network from the perspective of both users and providers. CaaS as an additional service (provided mostly in separate cache servers) gives the provider an opportunity to reduce both capital and operating costs using a fewer number of active physical machines for IaaS; and this can justify the cost of cache servers in our model. The user also benefits from CaaS in terms of application performance with minimal extra cost; besides, caching is enabled in a user transparent manner and cache capacity is not limited to local memory. The specific contributions of this paper are listed as follows: first, we design and implement an elastic cache system, as the architectural foundation of CaaS, with remote memory (RM) servers or solid state drives (SSDs); this system is designed to be pluggable and file system independent. By incorporating our software component in existing operating systems, we can configure various settings of storage hierarchies without any modification of operating systems and user applications. Currently, many users exploit memory of distributed machines (e.g., memcached) by integration of cache system and users' applications in an application level or a file-system level. In such cases, users or administrators should prepare cache-enabled versions for users' application or file system to deliver a cache benefit.

2. EXISTING SYSTEM

The past few years have seen the rise of smart handheld wireless devices as a means of content consumption. Content might include streaming applications in which chunks of the file must be received under hard delay constraints, as well as file downloads such as software updates that do not have such hard constraints. The core of the Internet is well provisioned, and network capacity constraints for content delivery are at the media vault (where content originates) and at the wireless access links at end-users. Hence, a natural location to place caches for a content distribution network (CDN) would be at the wireless gateway, which could be a cellular base station through which users obtain network access. Furthermore, it is natural to try to take advantage of the inherent broadcast nature of the wireless medium to satisfy multiple users simultaneously. There are multiple cellular base stations (BSs), each of which has a cache in which to store content. The content of the caches can be periodically refreshed through accessing a media vault. We divide users into different clusters, with the idea that all users in each cluster are geographically close such that they have statistically similar channel conditions and are able to access the same base stations. Note that multiple clusters could be present in the same cell based on the dissimilarity of their channel conditions to different base stations. The requests made by each cluster are aggregated at a logical entity that we call a front end (FE) associated with that cluster.

2.1. Disadvantages of Existing System

- The wireless network between the caches to the users has finite capacity.
- Refreshing content in the caches from the media vault incurs a cost.

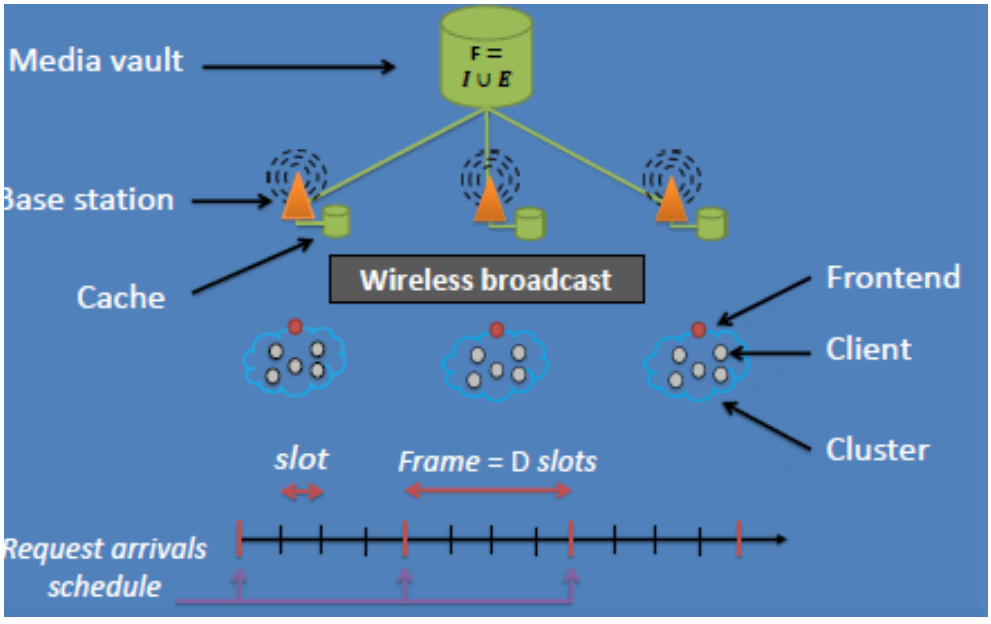
3. PROPOSED SYSTEM

In this paper, we develop algorithms for content distribution with elastic and inelastic requests. We use a request queue to implicitly determine the popularity of elastic content. Similarly, the deficit queue determines the necessary service for inelastic requests. Content may be refreshed periodically at caches. We study two different kinds of cost models, each of which is appropriate for a different content distribution scenario. The first is the case of file distribution (elastic) along with streaming of stored content (inelastic), where we model cost in terms of the frequency with which caches are refreshed. The second is the case of streaming of content that is generated in real-time, where content expires after a certain time, and the cost of placement of each packet in the cache is considered.

3.1. Advantages of Proposed System

- It stabilizes the system load within the capacity region.
- Minimizes the average expected cost while stabilizing the deficit queues

3.2. Proposed System Architecture



4. MODULES

- Creating System Model
- Content Caching System Module
- Elastic Traffic Module
- Inelastic Traffic Module

4.1. Creating System Model

- In this module, we create the System model, with Socket programming technique
- Create Wireless Nodes (Base Stations) with Cache
- Media Vault
- There are multiple cellular *base stations* (BSs), each of which has a cache in which to store content.
- Users can make two kinds of requests, namely: 1) elastic requests that have no delay constraints, and 2) inelastic requests that have a hard delay constraint.

4.2. Content Caching System Module

- In this module we design Scheduling methodology that is what is to be broadcasted from caches. In this module we also develop Content caching methodology, which is what to be loaded in caches.
- The content of the caches can be periodically refreshed through accessing a *media vault*. We divide users into different *clusters*, with the idea that all users in each cluster are geographically close such that they have statistically similar channel conditions and are able to access the same base stations. Note that multiple clusters could be present in the same cell based on the dissimilarity of their channel conditions to different base stations. The requests made by each cluster are aggregated at a logical entity that we call a *front end* (FE) associated with that cluster. The front end could be running on any of the devices in the cluster or at a base station, and its purpose is to keep track of the requests associated with the users of that cluster.

4.3. Elastic Traffic Module

- In this module, we develop elastic traffic module, where there should be No delay constraint.
- Stored in Request Queues at frontends.
- Elastic requests are stored in a *request queue* at each front end, with each type of request occupying a particular queue. Here, the objective is to stabilize the queue, so as to have finite delays.

4.4. Inelastic Traffic Module

- In this module, we develop inelastic traffic module for Hard Delay Constraint.
- Drop if not served by the deadline.
- Need a minimum delivery ratio.
- For inelastic requests, we adopt the model proposed wherein users request chunks of content that have a strict deadline, and the request is dropped if the deadline cannot be met.

4.5. Content Distribution Network System

In this module, we develop algorithms for content distribution with elastic and inelastic requests. We use a request queue to implicitly determine the popularity of elastic content. Similarly, the deficit queue determines the necessary service for inelastic requests. Content may be refreshed periodically at caches. We study two different kinds of cost models, each of which is appropriate for a different content distribution scenario. The first is the case of file distribution (elastic) along with streaming of stored content (inelastic), where we model cost in terms of the frequency with which caches are

refreshed. The second is the case of streaming of content that is generated in real-time, where content expires after a certain time, and the cost of placement of each packet in the cache is considered.

4.6. Content Caching System

In this module we design Scheduling methodology that is what is to be broadcasted from caches. The caches are all connected to a media vault that contains all the content. Users can often experience extended network access time and file downloading time due to poor Web document retrieval performance. Poor performance can occur because the WebSEAL server is waiting for documents retrieved from junctioned back-end servers. Caching of Web content gives you the flexibility of serving documents locally from Web SEAL rather than from a back-end server across a junction. The content caching feature allows you to store commonly accessed Web document types in the Web SEAL server's memory. Clients can experience much faster response to follow-up requests for documents that have been cached in the Web SEAL server. Cached content can include static text documents and graphic images. Dynamically generated documents, such as database query results, cannot be cached. Caching is performed on the basis of MIME type.

5. CONCLUSION AND FUTURE WORK

In this research work studied algorithms for content placement and scheduling in wireless broadcast networks. While there has been significant work on content caching algorithms, there is much less on the interaction of caching and networks. Converting the caching and load balancing problem into one of queuing and scheduling is hence interesting. Considered a system in which both inelastic and elastic requests coexist. Our objective was to stabilize the system in terms of finite queue lengths for elastic traffic and zero average deficit value for the inelastic traffic. In designing these schemes, showed that knowledge of the arrival process is of limited value to taking content placement decisions. Incorporated the cost of loading caches is in proposed problem with considering two different models. In the first model, cost corresponds to refreshing the caches with unit periodicity. In the second model relating to inelastic caching with expiry, directly assumed a unit cost for replacing each content after expiration. A max-weight-type policy was suggested for this model, which can stabilize the deficit queues and achieves an average cost that is arbitrarily close to the minimum cost.

REFERENCES

[1] N. Abedini and S. Shakkottai, "Content caching and scheduling in wireless broadcast networks with elastic and inelastic traffic," in *Proc.IEEE WiOpt*, 2011, pp. 125–132.

[2] I. Hou, V. Borkar, and P. Kumar, "A theory of QoS for wireless,"in *Proc. IEEE INFOCOM*, Rio de Janeiro, Brazil, Apr. 2009, pp.486–494.

[3] R. M. P. Raghavan, *Randomized Algorithms*. NewYork,NY,USA:Cambridge Univ. Press, 1995.

[4] P. Cao and S. Irani, "Cost-awareWWWproxy caching algorithms," in *Proc. USENIX Symp. Internet Technol. Syst.*, Berkeley, CA, Dec. 1997,p. 18.

[5] K. Psounis and B. Prabhakar, "Efficient randomized Web-cache replacement schemes using samples from past eviction times,"*IEEE/ACM Trans. Netw.*, vol. 10, no. 4, pp. 441–455, Aug. 2002.

[6] N. Laoutaris, O.T. Orestis, V.Zissimopoulos, and I. Stavrakakis, "Distributed selfish replication," *IEEE Trans. Parallel Distrib. Syst.*, vol.17, no. 12, pp. 1401–1413, Dec. 2006.

[7] S. Borst, V. Gupta, and A. Walid, "Distributed caching algorithms for content distribution networks," in *Proc. IEEE INFOCOM*, San Diego,CA, USA, Mar. 2010, pp. 1–9.

[8] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," *IEEE Trans. Autom. Control*, vol. 37, no.12, pp. 1936–1948, Dec. 1992.

[9] X. Lin and N. Shroff, "Joint rate control and scheduling in multihop wireless networks," in *Proc. 43rd IEEE CDC*, Paradise Islands, Bahamas,Dec. 2004, vol. 2, pp. 1484–1489.

[10] A. Stolyar, "Maximizing queueing network utility subject to stability:Greedy primal-dual algorithm," *Queueing Syst. Theory Appl.*, vol. 50,no. 4, pp. 401–457, 2005.

[11] A. Eryilmaz and R. Srikant, "Joint congestion control, routing, and MAC for stability and fairness in wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 8, pp. 1514–1524, Aug. 2006.

[12] J. Jaramillo and R. Srikant, "Optimal scheduling for fair resource allocation in ad hoc networks with elastic and inelastic traffic," in *Proc.IEEE INFOCOM*, San Diego, CA, USA, Mar. 2010, pp. 1–9.

[13] M. M. Amble, P. Parag, S. Shakkottai, and L. Ying, "Content-aware caching and traffic management in content distribution networks," in *Proc. IEEE INFOCOM*, Shanghai, China, Apr. 2011, pp. 2858–2866.

[14] M. Neely, "Energy optimal control for time-varying wireless networks," *IEEE Trans. Inf. Theory*, vol. 52, no. 7, pp. 2915–2934, Jul.2006.

[15] F. Foster, "On the stochastic matrices associated with certain queueing processes," *Ann. Math. Statist.*, vol. 24, pp. 355–360, 1953.

[16] M. Neely, "Energy optimal control for time varying wireless networks," *IEEE Trans. Inf. Theory*, vol. 52, no. 7, pp. 2915–2934, Jul.2006.