ISSN: 2321-7782 (Online)
Volume 3, Issue 7, July 2015
International Journal of Advance Research in Computer Science and Management Studies
Research Article / Survey Paper / Case Study
Available online at: www.ijarcsms.com

Content Caching and Scheduling in Wireless Networks with Elastic and Inelastic Traffic

V. Venkatesan[1]
Research Scholar,
Department of Computer Science,
H.H. The Rajahs College (Autonomous)
India

R. Malathi[2]
Assistant Professor,
Department of Computer Science,
H.H. The Rajahs College (Autonomous)
India

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.

Keywords: Content distribution network, Delay sensitive traffic, Quality of service, Prediction, Queuing.

I. 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.

II. Related Work

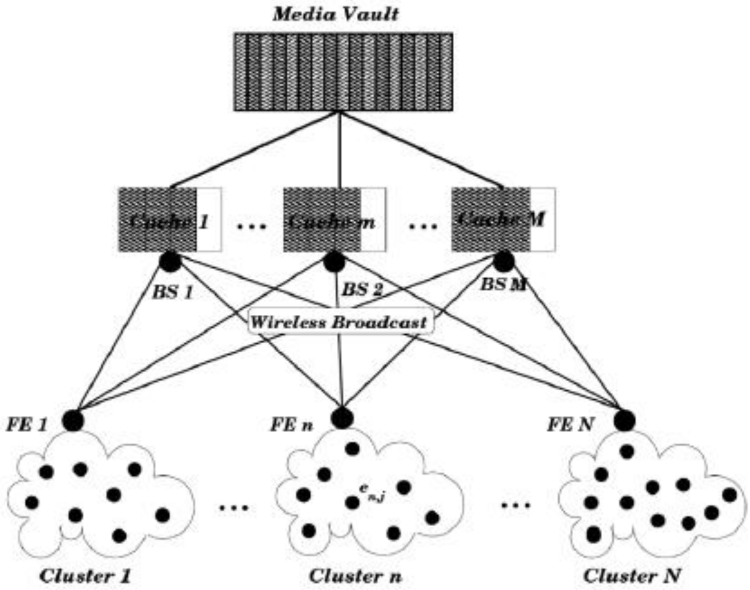
Jeffrey E. Wieselthier et al develop the Broadcast Incremental Power Algorithm, and adapt it to multicast operation as well. This algorithm exploits the broadcast nature of the wireless communication environment, and addresses the need for energy-efficient operation. we have identified some of the fundamental issues associated with energy-efficient broadcasting and multicasting in infrastructure less wireless networks, and we have presented preliminary algorithms for the solution of this problem. Our studies show that improved performance can be obtained when exploiting the properties of the wireless medium; i.e., networking schemes should reflect the node based nature of wireless communications, rather than simply adapt link-based schemes originally developed for wired networks. In particular, the Broadcast Incremental Power (BIP) Algorithm, which exploits the wireless multicast advantage, provides better performance than the other algorithms we have studied over a wide range of network examples. [1]

Meghana M Amble et al objective is to design policies for request routing, content placement and content eviction with the goal of small user delays. Stable policies ensure the finiteness of the request queues, while good polices also lead to short queue lengths. We first design a throughput-optimal algorithm that solves the routing-placement-eviction problem. The design yields insight into the impact of different cache refresh policies on queue length, and we construct throughput optimal algorithms that engender short queue lengths. We illustrate the potential of our approach through simulations on different CDN topologies.. Future work includes streaming traffic with requests that have hard delay constraints,and which are dropped if such a constraint cannot be met. [2]

Somanath Majhi et al Fourier series based curve fitting with the options of nonlinear least squares methodand trust-region algorithm is used to measure limit cycle parameters in thepresence of measurement noise. Examples are given to illustrate the valueof the proposed methodRelay feedback identification in process control can lead to erroneous results if the system parameters are estimated from the approximate describing function approach. Exact analytical expressions are derived and on the basis of these expressions an identification procedure is suggested which is capable of estimating the parameters of aclass of process transfer functions.When the limit cycle test measurements are error free, the accurate values of the model parameters areestimated. The solutions for the three examples show the generally accepted point about the DF method that its accuracy increases with therelative order of the plant transfer function which is successively 1, 2, and 5 in the examples. The present method can be applied to identifythe class of nonminimum phase processes in the presence of measurement noise.[3]

Bo Zhou et al formulate this stochastic optimization problem as an infinitehorizon average cost Markov decision process (MDP). It is wellknown to be a difficult problem and there generally only existnumerical solutions. By using relative value iteration algorithmand the special structures of the request queue dynamics,. we consider the optimal dynamic multicast scheduling to jointly minimize the average delay, power and fetching costs for cache-enabled content-centric wireless networks. We formulate this stochastic optimization problem as an infinite horizon average cost MDP. We show that the optimal policy has a switch structure in the uniform case anda partial switch structure in the non uniform case. Moreover, in the uniform case with two contents, we show that the switch curve is monotonically non-decreasing. The optimality properties obtained in this paper can provide design insights for multicast scheduling in practical cache-enabled content centric wireless networks [4].

III. METHODOLOGY USED



Client Module

This is any entity that requests data from a data server. When a client submits a request, besides the data it requests, it may also include some content service requirements, arising from device limitations and data format limitations

Intermediate Server Module

This is any entity that is allowed by a data server to provide content services in response to requests by clients. Intermediaries include caching proxies and transforming proxies. Totally we are using four proxy's. each proxy allotted for specific function. Based on user request the allotted proxy will process the request. if any one proxy busy with some function then the request delegated to the sub proxy that is assigned for the particular requested function.

Integrated Cache-Routing

- Nearest-caching tables can be used in conjunction with any underlying routing protocol to reach the nearest cache node, as long as the distances to other nodes are maintained by the routing protocol.
- However, note that maintaining cache-routing tables instead of nearest-cache tables *and* routing tables doesn't offer any clear advantage in terms of number of messages transmissions.
- We could maintain the integrated cache-routing tables in the similar vein as routing tables are maintained in mobile ad hoc networks. Alternatively, we could have the servers periodically broadcast the latest cache lists. In our simulations, we adopted the latter strategy, since it precludes the need to broadcast Add Cache and Delete Cache messages to some extent.

Localized Caching Policy

The caching policy of DCA is as follows. Each node computes benefit of data items based on its "local traffic" observed for a sufficiently long time. The *local traffic* of a node *i* includes its own local data requests, nonlocal data requests to data items cached at *i*, and the traffic that the node *i* is forwarding to other nodes in the network.

- A node decides to cache the most beneficial (in terms of local benefit per unit size of data item) data items that can fit in its local memory. When the local cache memory of a node is full, the following cache replacement policy is used.

- In particular, a data item is newly cached only if its local benefit is higher than the benefit threshold, and a data item replaces a set of cached data items only if the difference in their local benefits is greater than the benefit threshold.

Data Server Module

This is an entity that originally stores the data requested by a client. The data server module contains the Intermediary profile table. The table contains the detail about the proxy and specific function about particular proxy and their keys are all stored in that table. The server uses all these details when the client sends the request. And it contain function and corresponding privileges table also. This table contains the detail about what are the data present in the server and there privileges present in the table. The client request processed based on the privileges. It has two updates. Read and update.

IV. Proposed Work

In this research work, develop algorithms for content distribution with elastic and inelastic requests. 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.

Advantages of proposed system

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

V. Experimental Results

1. 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.

2. 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.

Algorithm Implementation

Decomposed Elastic-Inelastic Scheduling and Placement Scheme

Given the statistics of the requests and channel states, divide the available cache capacity v to v_E and $v - v_E$:

Elastic traffic:

Allocate v_E of the caches' capacity to elastic contents and use Algorithms 1 for service scheduling and content placement of the elastic requests.

Inelastic traffic:

At the beginning of frame k, given the deficit queue lengths, arrivals and the channel states,

let $d_u = d_u(k) + \sum_i a_{u,i}(k)$

Solve the following maximization problem to find the optimal inelastic schedule:

$$max \sum_{u,i} \{d_u\} + s_{u,i}$$

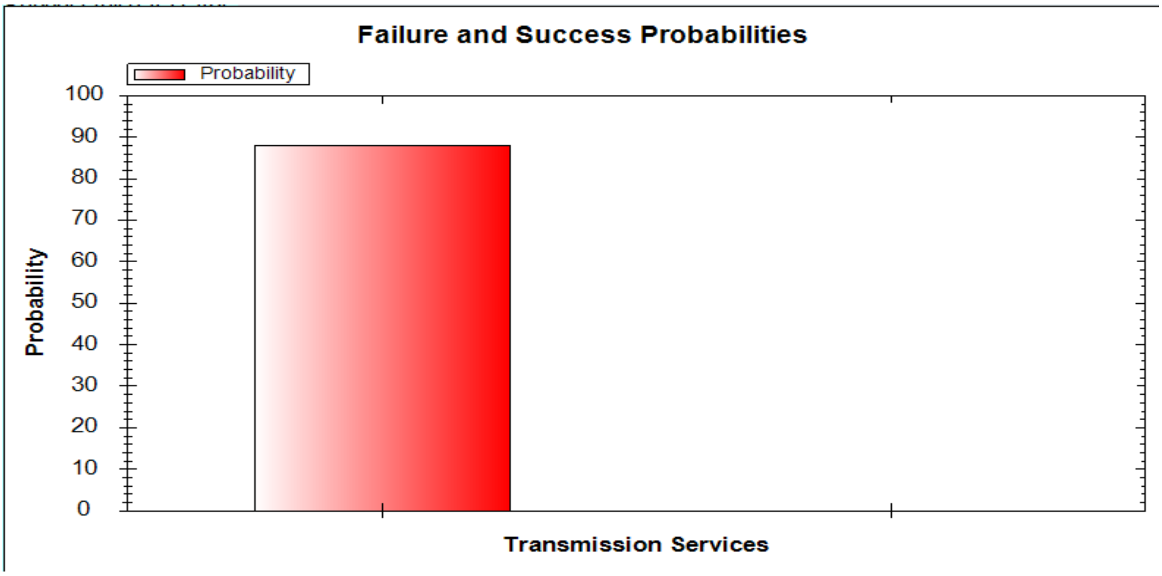
Subject to,

$s_{u,i} \le a_{u,i} \quad \forall_{u,i}$

$s_{u,i} \le \sum_m c_u^m \; s_i^m \quad \forall_{u,i}$

$\sum_i s_i^m \le \min(D, v - v_E) \quad \forall_m$

$s_i^m \in \{0,1\} \quad \forall_{m,i}$



	Packet Success Rate	
	Elastic Traffic (in Sec)	**Inelastic Traffic (in Sec)**
Energy Level Consumption	61.938	974.025
	61.5109	974.596
	61.441	974.478
	61.31	974.342
	61.128	974.179
	61.847	974.065
	499.5	974.944
	499.542	300.699
	499.473	300.767
	499.208	300.893

3. Elastic Traffic

In this module, we assume there are only requests for elastic content. These requests are to be served using unicast communications. For notational convenience, we assume that transmissions are between base stations and front ends, rather than to the actual users making the requests. We first determine the capacity region, which is the set of all feasible requests. Note that this model, in which front ends have independent and distinct channels to the caches, differs from the previously studied wired caching systems because the wireless channels are not always ON. Therefore, the placement and scheduling must be properly coordinated according to the channel states.

4. Inelastic Traffic

In this module, we focus an inelastic caching problem where the contents expire after some time. In this new model, which is compatible with real-time streaming of live events, we only consider inelastic traffic and assume that the lifetime of an inelastic content is equal to the length of a frame. Hence, we can cache a content only for the duration of a frame after which the content will not be useful any longer.

5. Joint elastic-inelastic scenario

In this section, we study the general case where elastic and inelastic requests coexist in the system. Recall that the elastic requests are assumed to be served through unicast communications between the caches and front ends, while the base stations broadcast the inelastic contents to the inelastic users. We further assumed servers can employ OFDMA method to simultaneously transmit over their single broadcast and multiple unicast channels. Although these two types of traffic do not share the access medium, all the content must share the common space in the caches. Consequently, we require an algorithm that jointly solves the elastic and inelastic scheduling problems. In this section, we first determine the general capacity region of the system, and then present our algorithm .

6. Pure unicast elastic scenario

In this section, we assume there are only requests for elastic content. As noted in the last section, these requests are to be served using unicast communications. For notational convenience, we assume that transmissions are between base stations and front ends, rather than to the actual users making the requests. We first determine the capacity region, which is the set of all feasible requests. Note that this model, in which front ends have independent and distinct channels to the caches, differs from the previously studied wired caching systems, because the wireless channels are not always ON. Therefore, the placement and scheduling must be properly coordinated according to the channel states.

VI. 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. On the Construction of Energy-Efficient Broadcast and Multicast Trees in Wireless Networks by Jeffrey E. Wieselthier ,Gam D. Nguyen , Anthony Ephremides.
2. Content-Aware Caching and Traffic Management in Content Distribution Networks by Meghana M Amble,, Parimal Parag, Srinivas Shakkottai , and Lei Ying.
3. Relay-Based Identification of a Class of NonminimumPhase SISO Processes by SomanathMajhiOptimal Dynamic Multicast Scheduling for Cache-Enabled Content-Centric Wireless Networks by Bo Zhou, Ying Cui and Meixia Tao.
4. Kaleidoscope: Cloud Micro-Elasticity via VM State Coloring by Roy Bryant, Alexey Tumanov ,Olga IrzakAdin Scannell, Kaustubh Joshi , Matti Hiltunen, H. Andr´es Lagar-Cavilla, Eyal de Lara.
5. True Elasticity in Multi-Tenant Data-Intensive ComputeClusters by Ganesh Ananthanarayanan,Christopher Douglas, Raghu Ramakrishnan.
6. I. Hou, V. Borkar, and P. Kumar, “A Theory of QoS for wireless,” in IEEE INFOCOM 2009, Rio de Janeiro, Brazil, April 2009.
7. R. M. P. Raghavan, Randomized Algorithms. New York: Cambridge University Press, 1995.

