Caching on steroids with Community Enabled Caching

Krishna Kant Singh
Department of
Computer Science and Engineering
Indian Institute of Technology
Email: cs15mtech11007@iith.ac.in

Kotaro Kataoka
Department of
Computer Science and Engineering
Indian Institute of Technology
Email: kotaro@iith.ac.in

***Abstract*—In-network caching is most essential part of Information Centric Networking. We explore a new caching mechanism based on the idea of Community Detection in a network made of edge routers and request patterns between them. Our work uses real word data from Yahoo Today Module Dataset, using this a resource request models is created for the user communities. Community detection algorithms are fundamental tools that allow us to uncover organizational principles in networks. When detecting communities, there are two possible sources of information one can use: the network structure, and the features and attributes of nodes. Community Detection seems a perfect match for managing the cache content at every node in Information Centric Networking where packets are cached at every node along the delivery path. Our work tries to utilize Community Detection for improving caching in the ICN framework. CENC creates a flow graph wherein nodes indicate the edge routers and the edges indicate interest packets that are common to both the nodes. Then using the Markov cluster algorithm non-overlapping communities are formed based on the flow graph. The algorithm uses preemptive caching hereafter, where if a content is cached at one of the edge routers r1 belonging to a community c1 then it is cached for all the other routers in the community as well. This paper argues through are experiments that this method complements already existing caching methods as well as it acts as a standalone caching scheme and it performs quite well. We evaluate the proposed CEN scheme using both theoretical tools and extensive simulations over real Internet domain topologies and compare their performance with and without are optimization. /*add more lines after evaluation*/**

***Index Terms*—Information Centric Networking , Markov Cluster Algorithm , Community Detection , Caching**

I. Introduction

Information-centric networking is a paradigm for the Internet which advocates ubiquitous in-network caching. It shifts from a host-centric view of the INTERNET to a content-centric view. In ICN every node that exists in the network can act as a cache source for the resources. This helps in reducing the latency of delivery of items, wastage of bandwidth for fetching duplicate items and most importantly the backbone of the network gets a lot less request due to this model. Every resource has unique name which helps in locating a resource and delivering it to the user. Caching in ICN presents us with both opportunists and challenges. On one hand caching the resources as close to the users as possible leads to reducing the delivery time of the resource, but on the other hand there is only a limited amount of space at each route. The problems of content placement in the caches, cache replacement policies that are suited for networking also come with its own challenges. Thus, using prior caching schemes used in OS are not well suited in case of communication networks.

Caching models in computer architecture talk about locality of reference principle which basically says that if a program at some time is accessing a part of a memory it in future will access the parts of memory near to it. A same analogy can be applied to caching in networking, our hypothesis centers upon that some videos are more popular in certain demographics than the others. So in different demographics it is seen that the request patters are very diffrent. Using global popularity of resources as discussed in previous papers would result in a high cache miss ratio. One of the major problems in using request patterns that vary with respect to demographic is modeling them. At present there is no accepted solution for that matter. Most vod caching system use traces for real network that are not readily available. Our work show how a widely used dataset the Yahoo Today Module can be leveraged for finding request pattern that mimic real world request patterns.

In our model every edge router serves as a reprsentative of mix demographics of varying degrees. We use mix of demographics so that our model generalizes well to real world scenarios. Our work introduces the concept of communities of edge routers in ICN. These communities can thought of as bunch of edge routers that have users with semantically similar interest and similar request patterns of resources. A pre-fetching scheme for caching of resources based on the community stucture is used. For the replacement policies at each router a cost-sensitive contextual armed bandit algorithm decides the which items to replace. For taking in to account changing popularity of content and modeling viral contnet. This is achived by introduction of high in-degree node at the resource graph.

Our present work combines both caching and community detection in the ICN architecture. Our work shows that it improves the caching performance of the existing caching techniques considerably. The main contribution of this paper are

- Framework that can be used with existing caching techniques like LRU,FIFO to boost their performance using community detection
- Giving the a reliable real world model for request patterns for resources that includes time variance of popularity of items. This request pattern can be used for analysis of any caching mechanism.
- Our caching scheme that uses 3 level architecture along with cost-sensitive contextual multi armed bandit for caching scheme outperforms existing caching schemes.
- Exploring the use of community structure in ICN. These can further explored be beneficial research in security in ICN.

The rest of the paper is organized as follows section II discuss previous work that is most relevant to this paper. Section III gives a detailed description of our CEN algorithm. Section IV shows the performance of our algorithm Section V concludes the paper and discusses future work.

II. Related Work

ICN differs from the traditional network in many ways, some of the most important distinctions are Caching, Unique naming of content,Request Model etc. Recently a lot of work has been done in these areas but particularly in the area of caching. The successful adoption of ICN is highly correlated with a powerful caching scheme. A good caching scheme can be defined as one in which the miss rate is a very low for the requested item and it has a better than linear performance improvement as the cache size increases.

Most of the previous work in the area of Caching in ICN deals with Independent request model scheme because of the inherent difficulty in modeling a dependent model for request of resources. Though some previous work like [4] looks into the problem using concepts like Hawkes process for modeling requests that are correlated in both time and space. But these methods are much more complex and not easy to comprehend, our approach is more intuitive and much simpler. We model traffic using creation of resource graph at each of the edge of the routers, then taking random walks to generate the requests. The inspiration of using such a model came from the recent work of Derbanch et al [5] where the model movie request patterns in large VOD systems using Gaussian distribution and using movie lens dataset for aggregating the requests. Their work though show that the request model used is very naive and has no real world characteristics about it unlike ours.

Caching popular content[3] or assigning probabilities to content and caching the one with the highest priority are well accepted techniques that are used at present. It can be shown that all these suffer from a very big flaw using just the global view of the data. This paper takes into account a 3-level scheme of popularity local popularity,community popularity and global popularity. These are discussed later in text.
Caching has been a topic of great intrest in ICN, but none of the previous work citeBernardini2013a[8][9] , to the best of our knowledge no work has explored inherit community structure that exists between the edge routers in the network or in general the INTERNET itself to enhance the performance of the caching strategies in ICN space. In any network ICN or IP based traditional network there exist a inherit community structure(similar content taste within a demographic region) between the users, this can be attributed to factors such as ethnicity, the social structure of society ,language and other factors . Exploiting this community structure in the context of caching can lead to better caching policies which are fine-tuned for a demographic region. The closest to our work is probably [7] which discusses the use of community detection for a better caching of web content in case of websites for better access time to users. Their work uses BGP routing logs for clustering of users in the traditional IP network . Our work differs in the context that clustering in the case of ICN network is nontrivial and also a much harder problem.

Detection of communities in the network can have far reaching implication not only for the caching but also in the areas of security in ICN. The work here explores how to find communities in an ICN framework and use the communities for optimizing pre-existing caching mechanisms such as LRU and FIFO.In community detection, the main goal is to cluster a group of vertices which have a dense connection(many edges) within them and the sparse connection with other vertices. Some algorithms take only the structure into account while others like CESNA[13] which also take node attributes into account. Most algorithms have an optimal function that they optimize to find the best communities eg modularity function. For formation of communites a very simple heuristic is followed ie if the 2 edge routers have the same content that have the same semantics then it can be assumed that the interest of users in the edge routers are similar. This simple heuristic can be shown to perform very well.

There exist basically 2 types of caching the On-Path Caching and Off-Path caching. In on-path caching the resources are cached everywhere along the path. In Off-Path Caching a hash function is computed based on the resource name and used to map it to a node number where it is cached. On-path caching provides you with the benefit of less delay but the trade off is that there is resource duplication in the network. Off-path caching there is no duplication of data, but at the cost of having higher delays. Hence there is a need for a mechanism that can have a balance between the two, duplication vs delay tradeoff is reduced. In our CENC algorithm tackles these problem by having caching at the edges only which in turn reduces the delay and also the algorithm only duplicates resource at another node when its assured the number of requests for that resource is greater than some threshold.

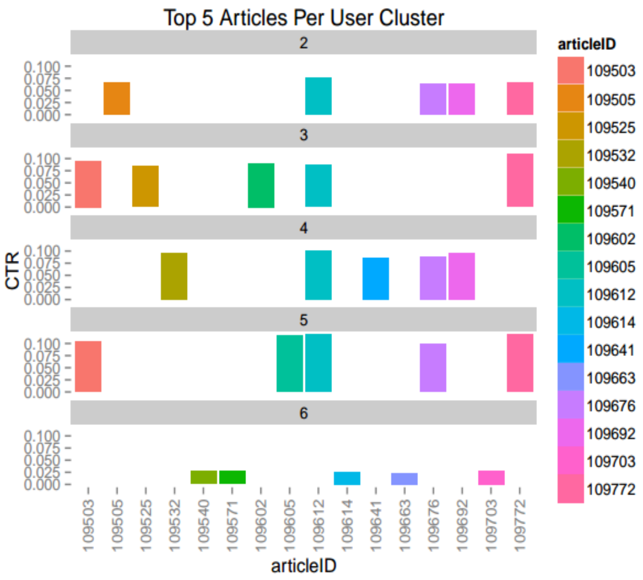

Fig. 1. Most Popular items in clusters

III. CEN:Community Enabled Caching

This section describes our model in detail. It starts with how the flow graph is created. The next section gives an overview of the clustering algorithm. Finally, to put it all together to show how the proposed method works.

A. Request Model

Algorithm 1 Request Model

```
1: procedure CREATE REQUEST GRAPH
2:     Input ← Log file of Yahoo Today Module, $\mathcal{P}_d$
3:     for $u_i$ in User do
4:         $C_i \leftarrow$ argmax $u_i$
5:         $j \leftarrow \mathcal{P}_d$
6:         $u_i$ belongs $e_j$
7:     for $E_i$ where $E_i \in EdgeRouter$ do:
8:         for $U_j$ where $U_j \in User$ do:
9:             G(V,$W_{ij}$)
10:            V: Set of Resources
11:            Let $P_i|P_j$ be the probability of resources i and j being togeher in the working set.
12:    Let $P_{ij}$ indicate $\frac{1}{W_{ij}}$
13:    Do a Markov Random walk for generating request for edge router.
```

The Yahoo Today module provides each user with 6 context vectors. It cluster the users with similar preference into one cluster. One of the methods that performs very well in practice for this dataset is to consider each context as a cluster, so user u_i belongs to cluster *argmax*u_i. A resource graph is created at each of these edge routers as G(V,W_{ij}) where V are set of resources and W_{ij} indicates the number of times r_i requested before r_j. Then using Markov random walk the requested are generated.

Lines 4-6 cluster the user using the context vector of the user. Hence the user is clustered into one of the 5 predefined clusters. The users from the cluster are then distributed using the user-given distribution of $\mathcal{P}_D$. Lines 7-11 create the resource graph. We look at the request made by the users in edge router e_j. These resources are treated as nodes of the graph and edges are weight according to $P(r_i|r_j)$ ie the probability of resources i and j being requested at edge router in the same time slice. Line 13 describes how are requests created using this resource graph.

B. Community Creation

Algorithm 2 Community Creation

```
procedure CREATE COMMUNITIES
    Input ← Resource Graph
    for $e_i \in EdgeRouter$ do
        for $e_j \in EdgeRouter$ do
            Find the similarity between $\phi(G_1)\phi(G_2)$
```

We propose this as a subgraph matching problem. Where using the Weisfiler-lehman kernel, find the the best matching subgraphs. For each of the edge routers and group them in one community. The wiesfiler-lehman kernel describes the frequency of the subgraphs in graph G_1. Weisfiler lehman kernel is given K=¡$\phi(G_1), \phi(G_2)$¿. Also the paper propose a extension to the weisfiler lehman kernel using the recently proposed word2vecmodel.

C. Opportunistic caching

Now, let us describe our caching algorithm. Create a 2-level cache for each routers.

- Level-1 or Local Cache- Prefetch all the k most popular items in for edge router i. These popularities are state stable probabilities that are gotten from the markov random walk.
- Level-2 or Commmunity Cache- Find a core router that is select as being sufficiently close to all the edge routers in the communities. This can be LCA of the edge routers in the case of tree-topology. The next k-most popular items prefecth in the communities.
- Oppurtunistic Caching- If in a edge router e_i of a community another item is gaining popularity. Then instead of caching this item at the local cache, cache this at the community cache level.

The cost sensitive CMAB perform the operation of replacing of content in the cache.

IV. Theoretical Improvements

The INTERNET traffic pattern has shown to follow a Zipf-like distribution [1]. Our simulations also model the traffic pattern as a Zipf-like distribution where the frequency of request for the item is dependent upon its rank. It can be show here how the number of misses in any common caching

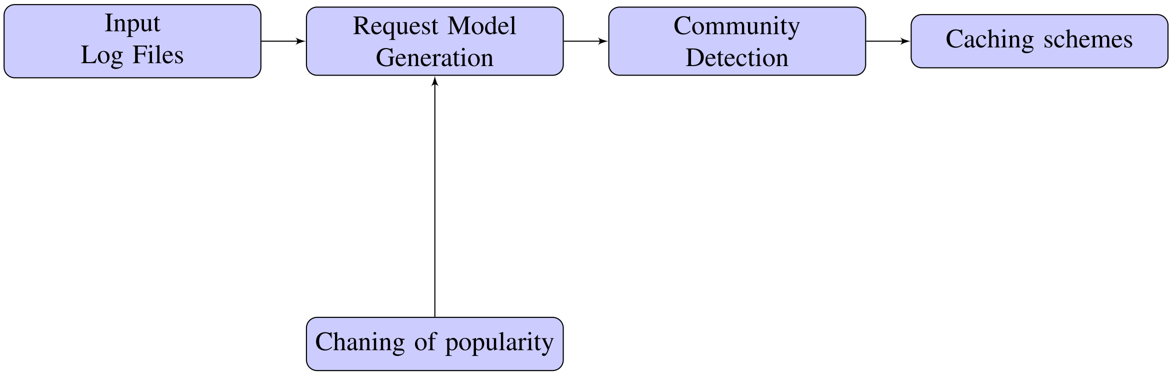

Fig. 2. 1.Log files consiting of user features and request pattern for uers. 2.Generaton of resource graph based on the input file where nodes are the resources and edges weights reprsent $P(R_i|R_j)$3.For imputing changing popularity and viral content. 4.Detection of communites based on requested patterns and semantic similarity of popular items.5.Prefecting caching scheme + CMAB for replacement of content trained online using the request pattern

schemes such as LRU,FIFO can be reduced by using vanilla CENC here the assumption is that there exist well-formed communities already and these communities are optimal. We give a lower bound on the number of such misses to show the minimum improvement that can be achieved using our caching scheme. Let X be a random variable,which represents the rank of the most frequently requested resource

Now compute the Expected number of requests from ER_1 for resources that already present in any other routers in community $\in$ C1.see Fig2

$$f(x) = c/Rank(x_i)^{\alpha} \quad wherei = 1, 2, ...R$$

$$c = \sum_{i=1}^{R} 1/i$$

Where R indicates all the resources in the universe and x_i denotes the rank of the resource i.α is a constant between 0 and 1

Let X_i denote a indicator random variable where

$$X_i = \begin{cases} 1 & \text{if } R_i \text{ was requsted by } er_j \text{ and if it was present in} \\ & \text{any of the } er_k \in C - er_j \\ 0 & \text{otherwise} \end{cases}$$

Now defining X = $\sum_{i=0}^{R} X_i$

$$E(X) = \sum_{i=0}^{R} E(X_i)$$

From linearity of expectation we get

$$E(X) = \sum_{i=0}^{R} [1.P(X_i = 1) + 0.P(X_i = 0)]$$

It is clearly visible that the X is sum of independent random variable , as request for R_i is independent of request for any other resource R_j

P(X_i) = P(X_i is requested by er_j) and (1 - P(X_i) is not present in any $er_k \in$ C -er_j)

Now using the zipf distribution,

$$P(X_i) = C/(x_i)^{\alpha} * (1 - C/(x_i)^{\alpha})^{(P-1)}$$

$$E(X) = \sum_{i=0}^{R} C/(x_i)^{\alpha} * (1 - C/(x_i)^{\alpha})^{(P-1)}$$

E(X) shows us the number of missed request ER_j would have to have to suffer if a simple caching mechanism such as LRU is used. Using chernoff bound the a lower limit on such cache misses is

$$P(X \leq (1-\delta)\mu) < \left(\frac{e^{-\delta}}{(1-\delta)^{1-\delta}} \right)^{\mu}$$

Here μ is equal to

$$\sum_{i=0}^{R} C/(x_i)^{\alpha} * (1 - C/(x_i)^{\alpha})^{(P-1)}$$

Using symmetry expand the argument for every router $\in$ C. Thus the expected number of misses that would occur when not using CEN is

$$Total\ misses_{withoutCEN} = E(X) * \sum |C'| : \ C' \geq 2$$

Hence it can be concluded that our algorithm performs O() given that we have been given these optimal communities.

V. Evaluation and Results

A. Traffic Modelling

Our biggest challenges lies in a way to generate a model of data that can be used to model traffic that captures the similarity between the traffic patterns.If we just use the IRM model with random sampling from the zipf distributed data then the communities are formed have not real common intrest that can be used to form communities. For solving this issue, in part the paper uses something similar to [5].It select a predefined number r for the number of communities formed then the cluster centers are sampled randomly from a $\mathcal{N}(0, \sigma_1)$ and then it generate data for each communities using $\mathcal{N}(0, \sigma_2)$ the variances σ_1 and σ_2 can be interpreted as inter and intra variances for the communities.A high value of σ_1 indicates that the inter-variance between communities ie. the intrest for the communites are quite varied,similarly a high value for $sigma_2$ indicate that within the communities the intrest are pretty varied.

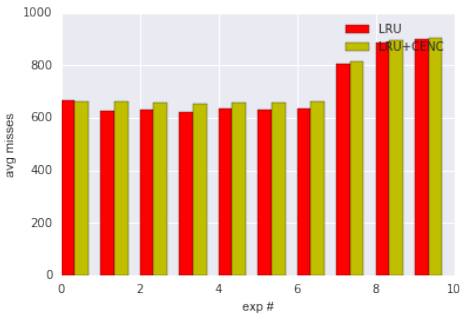

Fig. 3. Sample run on the radnomly sampled zipf data

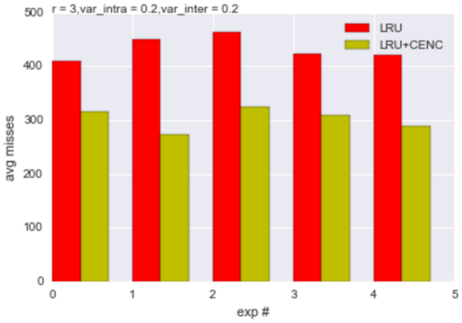

Fig. 4. Sample run on the gussian distributed data

B. Simulation Environment

Parmeters	Value	Parmeters	Value
Run/Simulation	10	Request Rate	50req/sec
Warmup Phase	True	chunck size	10Kb
Multipath	Disabled	caching replacement	CMAB
RoutingStrategy	Closest	Caching strategy	Pre-fect

We created our own simulation environment in python using simpy and networkx. The request for resources follow's a zipf madelbrot distribution with inter arrival time between requests by a poisson distribution.The topologies used are DEUTECHE TELECOM.Our main criteria for evaluation is cache miss ration with respect to cache size , diffrent values of $\sigma_1, \sigma_2 andr$ [9]. The simluation of ran for 10 experiments with a sample size of 600 packets,from the following figure it is quite clear that the LRU+CENC is performing worse than than the actual LRU algorithm this can be attributed to poor forming of the communities in the network since we use IRM model with random sampled data for each nodes there is no self similarity in traffic. But if use the traffic modelling techniques from Dernbach et al explained above it can be seen that LRU+CENC performs quite well than vanilla LRU. Then explore the relationship between r,σ_1,σ_2 using the following graphs.

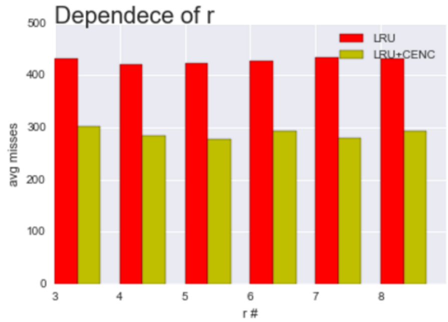

Fig. 5. Dependence on r

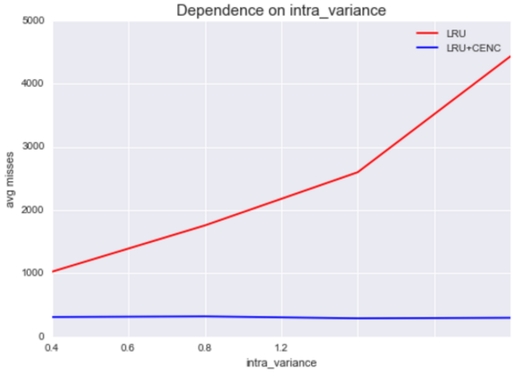

Fig. 6. Dependence on σ_2

References

[1] Lada a. Adamic and Bernardo a Huberman. Zipf's Law and the Internet. *Glottometrics*, 3:143–150, 2002.

[2] C??sar Bernardini, Thomas Silverston, and Olivier Festor. MPC: Popularity-based caching strategy for content centric networks. *IEEE International Conference on Communications*, pages 3619–3623, 2013.

[3] C??sar Bernardini, Thomas Silverston, and Olivier Festor. MPC: Popularity-based caching strategy for content centric networks. *IEEE International Conference on Communications*, pages 3619–3623, 2013.

[4] Ali Dabirmoghaddam, Maziar Mirzazad Barijough, and J J Garcia-Luna-Aceves. Understanding optimal caching and opportunistic caching at "the edge" of information-centric networks. *Proceedings of the 1st international conference on Information-centric networking - INC '14*, pages 47–56, 2014.

[5] Stefan Dernbach, Nina Taft, Jim Kurose, Udi Weinsberg, Christophe Diot, and Azin Ashkan. Cache Content-Selection Policies for Streaming Video Services. 2016.

[6] Yu Ke, Zhang Xinyu, Di Jiaxi, and Wu Xiaofei. Internet traffic identification based on community detection by label propagation. *Cloud Computing and Intelligent Systems (CCIS), 2012 IEEE 2nd International Conference on*, 02:786–791, 2012.

[7] Balachander Krishnamurthy and Jia Wang. On network-aware clustering of Web clients. *ACM SIGCOMM Computer Communication Review*, 30(4):97–110, 2000.

[8] Bighnaraj Panigrahi, Samar Shailendra, Hemant Kumar Rath, and Anantha Simha. Universal Caching Model and Markov-based cache analysis for Information Centric Networks. *Photonic Network Communications*, pages 1–6, 2015.

[9] Ioannis Psaras, Wei Koong Chai, and George Pavlou. Probabilistic in-network caching for information-centric networks. *Proceedings of the ICN workshop on information-centric networki*, pages 55–60, 2012.

[10] Usha Nandini Raghavan, R??ka Albert, and Soundar Kumara. Near linear time algorithm to detect community structures in large-scale networks. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, 76(3):1–12, 2007.

[11] Martin Rosvall, Alcides V. Esquivel, Andrea Lancichinetti, Jevin D. West, and Renaud Lambiotte. Memory in network flows and its effects on spreading dynamics and community detection. *Nature communications*, 5:4630, 2014.

[12] Stjin van Dongen. Graph clustering. *Graph stimulation by flow clustering*, PhD thesis:University of Utrecht, 2000.

[13] Yuan Zhang, Elizaveta Levina, and Ji Zhu. Community Detection in Networks with Node Features. *arXiv preprint arXiv:1509.01173*, pages 1–16, 2015.