

基于内容热度与节点介数的 NDN 网络缓存策略^①

郭晨, 郑焯, 丁尧, 王嵩

(中国科学技术大学 自动化系 未来网络实验室, 合肥 230026)

摘要: 缓存技术是数据命名网络 (Named data networking, NDN) 的关键技术之一. NDN 传统的 LCE 缓存策略会造成较大的冗余. 改进的 RCOne 策略采用随机放置的方法, 没有利用任何内容、节点信息, 对网络缓存性能的提升有限. Betw 策略只考虑到节点介数, 导致高介数节点缓存更替频繁, 当节点缓存容量远小于内容总量时, 缓存性能下降. 为了解决这些问题, 本文提出一种结合内容热度与节点介数的新型缓存策略 HotBetw (Hot content placed on node with high Betweenness), 充分利用内容与节点信息选择最佳的位置放置缓存. 仿真实验表明相对于典型 NDN 缓存策略, HotBetw 缓存策略在提高缓存命中率、降低平均跳数方面具有很好的效果.

关键词: 数据命名网络; 缓存技术; 内容热度; 节点介数; HotBetw 缓存策略

引用格式: 郭晨, 郑焯, 丁尧, 王嵩. 基于内容热度与节点介数的 NDN 网络缓存策略. 计算机系统应用, 2017, 26(12): 165-169. <http://www.c-s-a.org.cn/1003-3254/6105.html>

Cache Scheme Based on Content Popularity and Node Betweenness in Named Data Networking

GUO Chen, ZHENG Quan, DING Yao, WANG Song

(Laboratory for Future Networks, Department of Automation, University of Science and Technology of China, Hefei 230026, China)

Abstract: Cache is one of the key technologies of named data networking (NDN). However, the basic cache scheme LCE (leave copy everywhere) in NDN leads to much redundancy. The RCOne scheme chooses the cache node randomly without using any information of content and node, which is limited in improving cache performance. The Betw scheme results in that the node has the more frequent replacement with the larger betweenness centrality, which will decrease the cache performance when the node's cache capacity is far smaller than the total content amount. In order to solve those problems, a cache scheme named HotBetw is proposed in this paper based on content popularity and node betweenness to choose appropriate cache node along the content delivery path. The simulation results show that the HotBetw cache scheme can achieve higher cache hit ratio and reduce average request hop compared with existing schemes.

Key words: named data networking; cache; content popularity; betweenness centrality; HotBetw cache scheme

为了从根本上解决传统 IP 网络传输效率低的问题, 一类以信息为中心的新型网络体系——ICN 被提出^[1]. 数据命名网络 NDN 是 ICN 的主流网络架构之一^[2], NDN 网络泛在化的缓存能有效的降低用户的等待时延、节约带宽资源、减小服务器负载.

NDN 网络默认的缓存决策策略是 LCE (Leave copy everywhere)^[1], LCE 策略会在网络中产生大量冗余副

本, 导致网络缓存资源的利用率低. 针对 LCE 的缺点, RCOne^[3]、Betw^[4]等缓存决策策略被提出^[5,6]. 为了提升网络缓存的多样性, RCOne 算法在内容返回路径中随机地选择一个节点缓存内容. 这种随机化的缓存决策易于实现, 但并未考虑内容的热度等信息, 对缓存性能的提升有限. Betw 算法在内容返回时选择兴趣包请求路径上最重要的节点缓存, 其它节点不再缓存. 对于不同

① 收稿时间: 2017-03-14; 修改时间: 2017-03-31; 采用时间: 2017-04-07

网络拓扑, 该策略都取得了较高的网内节点缓存命中率, 并减少了内容传输的平均跳数. 然而, 在实际网络中, 节点缓存量远小于内容总量, Betw 策略会导致节点越重要, 到达的请求越多, 需要缓存的内容也越多, 同时节点负载也会越大, 从而导致缓存中的内容更替频繁, 新缓存的内容, 即使具有很高的热度, 也具有较大可能性被快速替换掉, 致使后续请求无法充分利用前期缓存.

基于以上方案存在的问题, 本文提出一种综合考虑内容热度与节点介数的网络缓存决策策略 HotBetw. 根据内容的热度信息将内容分为热点内容与非热点内容. 在内容返回的过程中结合内容热度与节点的重要性, 将热点内容放置在重要节点上, 非热点内容随机放置在某一个节点上. 该策略既能保证热点内容尽量缓存在相对重要节点上, 又能避免重要节点处于高频率的内容替换状态下, 同时也充分利用了非重要节点的缓存资源, 达到提升网络缓存性能的目的.

本文的结构如下: 第1节介绍和分析内容热度算法与 NDN 缓存决策策略的相关技术; 第2节阐述本文提出的 HotBetw 缓存决策策略; 第3节给出仿真环境、方案与参数的设置, 并分析仿真结果; 第4节对本文工作进行总结.

1 相关技术

本文旨在结合内容热度与节点介数进行缓存决策, 因此接下来将详细介绍内容热度算法、NDN 缓存决策策略这两方面的相关技术.

1.1 内容热度算法

为了提高缓存命中率, 节点进行缓存替换时会把热度最小的内容替换出去. 最常用的 NDN 网络缓存替换算法有 LRU^[7]与 LFU^[8]. LRU(Least recently used) 主要思想是: 当有新内容转发到节点且节点缓存空间已满时, 将最久未使用的缓存内容替换掉. LRU 主要考虑内容的动态性, 认为最近转发的内容很有可能再次被请求, 因此具有较大的热度. 但是如果一个内容每隔一段时间都会被请求一次, LRU 算法会用最近的内容将它替换掉, 即使最近的内容很少有人感兴趣. LRU 算法只注重了内容的动态性却忽略了内容的历史热度. LFU(Least frequently used) 的核心思想是给每一个缓存内容进行计数, 缓存命中时, 计数值加 1. 缓存替换时将计数值最小的内容替换掉. LFU 主要考虑内容的历史热度, 认为历史热度高的内容被再次请求的概率大. 但是如果请求的内容在动态变化时, LFU 算法的性能将下降. 原因是: 当一个内容过去很流行, 即使之后没

有人再请求, 它也会一直留在缓存中直到最近的一些被命中次数更多的内容出现. LFU 算法只注重了内容的历史热度却忽略了内容的动态性.

针对 LRU 与 LFU 缓存替换策略的不足, 文献[8]提出了 CCP 缓存替换策略. CCP 缓存替换策略周期性地统计内容的命中次数, 实时计算内容的热度, 根据内容热度进行缓存替换. 内容热度的计算公式如下:

$$P[i+1] = \beta * P[i] + (1 - \beta) * N[i] \quad (1)$$

其中, i 是计数周期, $P[i]$ 是内容在第 i 个周期的内容热度值, $N[i]$ 是该内容在第 i 个周期的缓存命中次数, β 是衰减因子, $0 < \beta < 1$, 表示上一个周期的热度值在当前周期所占的比例. 通过上述递推公式计算当前的内容热度, 既考虑了之前的热度对现在的影响, 更突出了最新的网络热点的影响力. CCP 兼顾了内容的动态性与历史热度. 本文将采用 CCP 缓存替换策略, 节点实时计算内容热度, 根据热度值, 将内容划分为热点内容与非热点内容.

1.2 NDN 缓存决策策略

NDN 的默认缓存决策策略是 LCE(Leave copy everywhere). LCE 的主要思想是在内容返回过程中的每一个节点都缓存内容. LCE 的算法复杂度低、实现简单, 但是会导致网络中出现大量冗余副本, 降低缓存的多样性.

RConc 缓存决策策略采用随机化的思想, 在内容返回路径中随机选取一个节点缓存内容. 图 1 所示的拓扑结构中, 用户 Client A 请求内容 c_1 时, 源服务器 s_1 提供 c_1 , c_1 在返回过程中随机选择 v_1 、 v_2 、 v_3 中的某一个节点缓存下来. 相比于 LCE, RConc 虽然能在一定程度上提升网络缓存多样性, 但是在缓存放置时既没有考虑内容热度, 也没有考虑节点信息, 对网络缓存性能的提升有限.

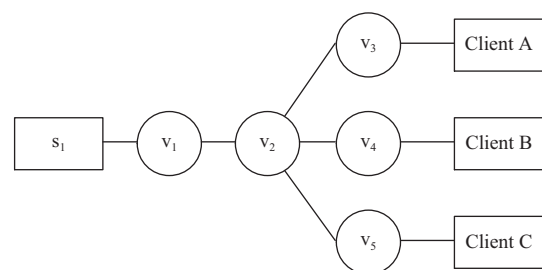


图 1 实例拓扑

文献[4]提出的 Betw 缓存决策策略, 请求命中后将内容只缓存在内容返回路径中最重要节点上. 由于在重要节点上, 同一个内容被请求的次数会更多, 使得

内容在节点上被缓存时间变长,缓存命中率变高从而提升缓存性能.而节点重要程度是以节点介数作为度量,节点介数用来描述节点在网络中重要性的一个度量,源于复杂网络领域,其数学定义如下:

$$C_B(v) = \sum_{s \neq v, t \in V} \frac{\sigma_{s,t}(v)}{\sigma_{s,t}} \quad (2)$$

其中 V 是网络拓扑结构中节点的集合, s 、 t 、 v 是拓扑结构中互不相同的 3 个节点, $C_B(v)$ 是节点 v 的节点介数, $\sigma_{s,t}$ 是节点 s 到节点 v 的所有最短路径数, $\sigma_{s,t}(v)$ 是其中经过节点 v 的最短路径数. 节点的介数越大,它在网络中越重要. 图 1 的拓扑结构中 v_2 的节点介数最大, 3 个用户节点请求都会经过 v_2 , Betw 策略将内容缓存在 v_2 节点可以满足更多的请求.

然而在实际网络中节点缓存大小与网络内容总量相比非常小,因此,重要节点到达的请求数量非常大,节点处于高频率的内容替换状态下,导致热点内容也会出现很快就被替换.此时大量的兴趣包到达重要节点后,无法实现缓存命中,兴趣包也只能被转发到别的节点,直至内容源端.在图 1 的拓扑结构中, Betw 策略会把所有的内容都缓存在 v_2 节点, v_2 节点的缓存空间大小有限,高频率的替换状态下导致网络缓存性能的降低;此外 v_2 之外的其他节点均未放置缓存,也造成了缓存资源的浪费.

2 算法介绍

2.1 HotBetw 缓存决策策略核心思想

基于以上分析,我们发现如果将内容缓存在内容返回路径中最重要的节点上,虽然能够满足更多的请求,但是由于节点缓存空间有限,缓存替换过快会导致网络缓存性能下降.另一方面,如果采用随机缓存的方法,由于没有结合任何内容或者节点的信息,网络缓存性能的提升也很有限,因此本文提出如下基于内容热度与节点信息的 HotBetw 缓存决策策略:

1) 根据内容热度算法,将内容分为热点内容与非热点内容.

2) 热点内容采用基于节点介数的 Betw 策略,缓存在内容返回路径中节点介数最大的节点上.

3) 非热点内容采用随机化的 RCOne 策略,随机缓存在内容返回路径中的某一个节点上.

下面通过一个实例来分析 HotBetw 策略的性能.在图 1 所示的拓扑结构中,假设每个节点的 CS 缓存表的大小为 1, 3 个用户节点分别先后请求 1 次内容: Client A 请求 c_1 , 然后 Client B 请求 c_2 , 最后 Client C 请

求 c_1 . 接下来我们分析采用不同缓存决策策略下,网络的缓存命中次数.缓存决策策略采用 Betw 策略时, Client A 请求的 c_1 由源服务器 s_1 提供, c_1 缓存在 v_2 节点; Client B 请求的 c_2 由源服务器 s_1 提供, c_2 缓存在 v_2 节点,将之前的 c_1 替换掉;最后 Client C 请求的 c_1 依然不能被缓存命中,只能由源服务器 s_1 提供. 3 次请求过后,缓存命中次数为 0. 如果缓存决策策略采用 RCOne, 当 Client A 请求的 c_1 缓存在 v_1 节点, Client B 请求的 c_2 缓存在 v_2 或 v_4 节点;或者 Client A 请求的 c_1 缓存在 v_2 节点, Client B 请求的 c_2 缓存在 v_1 或 v_4 节点时, Client C 请求的 c_1 才会被缓存命中. 可以算出 RCOne 策略的平均缓存命中次数为 4/9. 如果缓存决策策略采用 HotBetw, c_1 被请求 2 次,是热点内容,采用 Betw 策略; c_2 只被请求 1 次,是非热点内容,采用 RCOne 策略. Client A 请求的 c_1 由源服务器 s_1 提供, c_1 缓存在 v_2 节点; Client B 请求的 c_2 由源服务器 s_1 提供, c_2 只要不缓存在 v_2 节点, Client C 请求的 c_1 就会被 v_2 节点的缓存命中. 可以计算出 HotBetw 策略的平均缓存命中次数为 2/3, 高于 Betw 策略与 RCOne 策略. HotBetw 策略既能将热点内容放置在重要节点,又能降低重要节点缓存替换率提高缓存命中率,同时利用其它节点存储非热点内容,提升网络缓存多样性.

2.2 基于 CCP 缓存替换策略的内容热度算法

合理划分热点内容与非热点内容是 HotBetw 策略的关键之处.在实际网络中,我们无法像 2.1 小节实例中那样提前预知具体的内容请求,此时正确计算内容的实时热度显得尤为关键.文献[8]中提出的 CCP 缓存替换策略中能兼顾内容请求的动态变化与内容的历史热度,正确计算出内容的实时热度,因此本文采用 CCP 缓存替换策略所提出的内容热度算法计算内容热度. CCP 替换策略在 NDN 网络的 CS、PIT 以及 FIB 结构的基础上,新增了内容热度统计表(Content popularity table, CPT). CPT 表用来保存缓存内容的命中次数、历史热度用来计算缓存内容的实时热度.当 CS 表更新时, CPT 表也同步更新,确保 CPT 表保存 CS 表中所有缓存内容的信息. CPT 表详细结构如表 1 所示.

CPT 表保存的缓存内容信息主要包含以下 4 项:内容名称、历史热度、当前周期命中次数、实时热度. 当一个计时周期结束后,根据内容的历史热度、当前周期命中次数,代入 1.2 小节的公式 1 就可以计算出内容的实时热度,根据内容的实时热度进行缓存替换. 表 1 所示数据是衰减因子 $\beta = 0.4$ 时,依据公式 1 计算出的实时热度.

表1 内容热度统计表(CPT)

内容名称	历史热度	当前周期命中次数	实时热度
/content/1.mpg	12.41	20	16.96
/content/2.mpg	19.96	11	14.58
/content/3.mpg	23.41	32	28.56

根据内容的实时热度值,我们可以对内容进行合理的划分. NDN网络中用户请求到的内容是由源服务器节点或者请求路由路径中某个节点的CS表提供^[9]. 由源服务器节点提供的内容在请求路由路径中没有缓存副本,说明这个内容在路径中所有节点上的热度都不高,因此被划分为非热点内容. 由节点CS缓存表提供的内容根据实时热度值进行划分. 内容分类的详细信息如下:

1) 热点内容: CS表中实时热度值位于前40%的内容.

2) 非热点内容: 源服务器节点提供的内容、节点CS表实时热度值位于后60%的内容.

兴趣包请求如果被某一个节点CS表中实时热度值位于前40%的内容缓存命中,那么返回的内容是热点内容;如果被源服务器或者某一个节点CS表中实时热度值位于后60%的内容满足,则返回的内容是非热点内容.

2.3 HotBetw 缓存决策策略的实现

为了实现 HotBetw 缓存决策策略,兴趣包在请求的过程中,需要记录路由路径中所有节点的最大节点介数 C_B 和跳数 hop. 兴趣包被满足时,基于 2.2 小节的内容热度算法,判断内容是否为热点内容. 热点内容采用 Betw 缓存决策策略,在内容返回的路径中,将路径中每个节点 v 的节点介数 $C_B(v)$ 与最大节点介数 C_B 进行比较,两个值相同则将内容缓存在这个节点 v 上. 非热点内容采用 ROne 缓存决策策略,兴趣包被满足时产生一个 0 到跳数 hop 之间的随机数,内容返回过程中每转发 1 跳,随机数减 1,当随机数减为 0 时,将内容缓存在这个节点上. 图 2 的伪代码给出了节点对兴趣包与内容的详细处理过程. 兴趣包到达节点 v 时, OnIncomingInterest 首先查询兴趣包是否被节点 v 满足 (v 的 CS 表缓存命中或者 v 是源服务器 server). 如果被满足,则返回内容,内容从兴趣包中获得 3 项信息: 最大节点介数 C_B 、0-hop 之间的一个随机数,热点内容标志位 hot. 如果兴趣包没被满足,则更新兴趣包的最大节点介数 C_B 与跳数 hop,继续转发到下一跳. 内容到达节点 v 时, OnIncomingData 首先判断内容是否为热点内容,热点内容根据节点介数的匹配结果放置,非热点内容则根据跳数值是否为 0 进行放置,更新跳数值,继续将内容转发到下一跳.

```

Initialize (  $C_B = 0$ , hop=0, Bool hot=false )
OnIncomingInterest in node v:
    get(hop,hot);
    hop++;
    if data in cache
        hot=CPT.isHoted(data); //look up the CPT table
        hop=random(0,hop); //RCOne
        send(data);
    else if node v is server
        hot=false;
        hop=random(0,hop);
        send(data);
    else
        get( $C_B$ )
        if  $C_B(v) > C_B$ 
             $C_B = C_B(v)$ 
        end if
        forward interest to the next hop
    end if
OnIncomingData in node v:
    get(  $C_B$ ,hop,hot);
    hop--;
    if hot==true
        if  $C_B(v) == C_B$ 
            cache(data);
        end if
    else
        if hop==0
            cache(data);
        end if
    end if
    forward data to the next hop;

```

图2 HotBetw 伪代码

3 实验分析

3.1 实验环境与参数配置

本文在 Ubuntu 环境下,采用基于 NS-3 的 ndn SIM2.2^[10]进行仿真. 仿真拓扑结构采用树状拓扑结构,深度为 8,共有 50 个节点,其中 25 个叶子节点,1 个根节点,叶子节点是用户节点,根节点是源服务器节点. 源服务器提供 $N = 2000$ 种不同的内容. 每种内容的大小相同,内容的流行度服从 Zipf-Mandelbrot 分布. 每个节点缓存容量为 C , C 设置在 5-60 之间,使 CS 表缓存大小与内容总量之比 R 控制在 0.0025~0.030 之间,确保节点缓存容量远小于内容总量以符合网络真实情况,其中 $R = C/N$. 每个用户节点的请求频率都服从泊松分布,参数 $\lambda = 100 \text{req/s}$. 缓存替换策略采用 CCP 替换策略,

衰减因子 β 为0.4, 替换周期 T 为1s, 仿真时长为100s. 性能指标采用全网缓存命中率与请求平均跳数. 在缓存替换策略都采用CCP替换策略的情况下, 节点缓存容量与内容总量之比 R 不同时, 对比LCE、RCOne、Betw、HotBetw这4种缓存决策策略下网络缓存的性能.

3.2 实验结果分析

图3是在4种缓存决策策略下, 节点缓存容量不同时, 用户请求在全网的缓存命中率. 这个命中率是在节点缓存大小确定好后, 一次实验中被网内节点缓存命中的请求占所有用户请求的比例. LCE、RCOne、Betw这3组对照实验中, Betw策略的全网缓存命中率最高. 本文提出的HotBetw策略的全网缓存命中率相比于Betw策略高出9%.

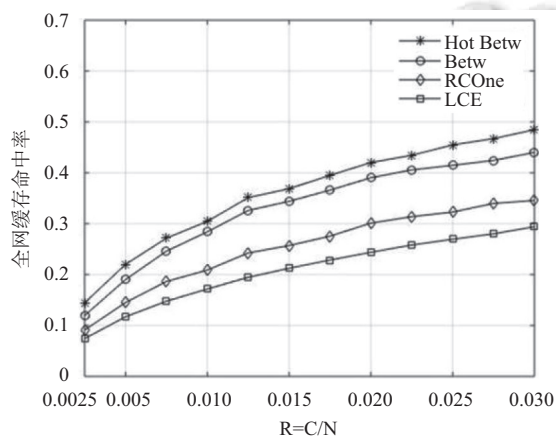


图3 全网缓存命中率随节点缓存大小的变化

图4是在4种缓存决策策略下, 请求的平均跳数随节点缓存容量大小的变化. 平均跳数是一次实验中所有的用户请求被转发的平均跳数. 3组对照实验中, Betw策略的平均跳数最低. 本文提出的HotBetw策略的平均跳数比Betw策略低0.3跳, 性能提升了10%.

4 结语

为了解决NDN网络中内容在请求路径上如何选择缓存节点的问题, 本文提出了一种基于内容热度与节点介数的缓存决策策略HotBetw. 该策略通过内容的热度信息, 对内容进行分类, 热点内容采用Betw策略, 非热点内容采用RCOne策略. HotBetw策略既能保证热点内容尽量缓存在相对重要节点上, 又能避免Betw策略只考虑节点介数而导致重要节点内容替换频繁, 同时也充分利用了非重要节点的缓存资源. 仿真实验表明, HotBetw决策策略与Betw策略相比全网命中率提升了9%, 请求平均跳数减小了10%, 相比于随机放置的

RCOne与到处放置的LCE策略性能提高更加显著.

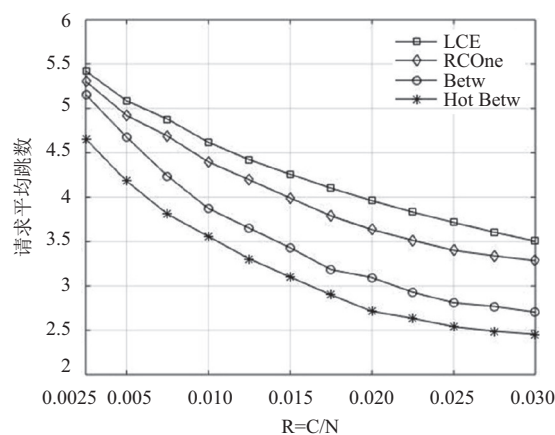


图4 请求平均跳数随节点缓存大小的变化

参考文献

- 1 张国强, 李杨, 林涛, 等. 信息中心网络中的内置缓存技术研究. 软件学报, 2014, 25(1): 154-175. [doi: 10.13328/j.cnki.jos.004494]
- 2 Ahlgren B, Dannewitz C, Imbrenda C, et al. A survey of information-centric networking. IEEE Communications Magazine, 2012, 50(7): 26-36. [doi: 10.1109/MCOM.2012.6231276]
- 3 Eum S, Nakauchi K, Murata M, et al. CATT: Potential based routing with content caching for ICN. Proc. of the Second Edition of the ICN Workshop on Information-Centric Networking. New York, USA. 2012. 49-54.
- 4 Chai WK, He DL, Psaras I, et al. Cache "less for more" in information-centric networks. International Conference on Research in Networking. Berlin Heidelberg. 2012. 27-40.
- 5 Psaras I, Chai WK, Pavlou G. Probabilistic in-network caching for information-centric networks. Proc. of the Second Edition of the ICN Workshop on Information-Centric Networking. New York, USA. 2012. 55-60.
- 6 Laoutaris N, Che H, Stavrakakis I. The LCD interconnection of LRU caches and its analysis. Performance Evaluation, 2006, 63(7): 609-634. [doi: 10.1016/j.peva.2005.05.003]
- 7 Podlipnig S, Böszörményi L. A survey of web cache replacement strategies. ACM Computing Surveys (CSUR), 2003, 35(4): 374-398. [doi: 10.1145/954339]
- 8 Ran JH, Lv N, Zhang D, et al. On performance of cache policies in named data networking. Proc. of the 2013 International Conference on Advanced Computer Science and Electronics Information. Beijing, China. 2013. 668-671.
- 9 Yi C, Afanasyev A, Wang L, et al. Adaptive forwarding in named data networking. ACM SIGCOMM Computer Communication Review, 2012, 42(3): 62-67. [doi: 10.1145/2317307]
- 10 Mastorakis S, Afanasyev A, Moiseenko I, et al. ndnSIM 2.0: A new version of the NDN simulator for NS-3 [Technical Report]. NDN-0028. Los Angeles: NDN, 2015.