

# 一种基于两级缓存的协同缓存机制\*

刘嘉琦<sup>1,2</sup>, 张亚文<sup>1,2</sup>, 张瀚文<sup>1,2</sup>, 孟绪颖<sup>1,2</sup>, 周继华<sup>3</sup>, 张玉军<sup>1,2</sup>

<sup>1</sup>(中国科学院 计算技术研究所, 北京 100190)

<sup>2</sup>(中国科学院大学, 北京 100190)

<sup>3</sup>(金美通信, 重庆 400030)

通讯作者: 张瀚文, E-mail: hwzhang@ict.ac.cn



**摘要:** 信息中心网络(information-centric networking, 简称 ICN)将网络通信模式从当前的以地址为中心转变为以信息为中心。泛在化缓存是 ICN 重要特性之一, 它通过赋予网络任意节点缓存的能力来缓和服务器的压力, 降低用户访问延迟。然而, 由于缺少内容热度的分布感知, 现有 ICN 缓存策略仍存在缓存利用率较低、缓存位置缺乏合理规划等问题。为了解决这些问题, 提出一种基于两级缓存的协同缓存机制(a cache coordination scheme based on two-level cache, 简称 CSTC)。将每个节点的缓存空间分为热度感知和协作分配两部分, 为不同热度的内容提供不同的缓存策略。同时, 结合提出的热度筛选机制和路由策略, 降低了缓存冗余, 实现了缓存位置优化。最后, 基于真实网络拓扑的仿真实验表明, CSTC 在次热门内容缓存数量上提升了 2 倍, 缓存命中率提升了将近 50%, 且平均往返跳数在多数情况下优于现有 On-path 缓存方式。

**关键词:** 信息中心网络; 内容中心网络; 网络化缓存; 缓存; 协同缓存

**中图法分类号:** TP303

中文引用格式: 刘嘉琦, 张亚文, 张瀚文, 孟绪颖, 周继华, 张玉军. 一种基于两级缓存的协同缓存机制. 软件学报, 2021, 32(9): 2963–2976. <http://www.jos.org.cn/1000-9825/5981.htm>

英文引用格式: Liu JQ, Zhang YW, Zhang HW, Meng XY, Zhou JH, Zhang YJ. Cache coordination scheme based on two-level cache. Ruan Jian Xue Bao/Journal of Software, 2021, 32(9): 2963–2976 (in Chinese). <http://www.jos.org.cn/1000-9825/5981.htm>

## Cache Coordination Scheme Based on Two-level Cache

LIU Jia-Qi<sup>1,2</sup>, ZHANG Ya-Wen<sup>1,2</sup>, ZHANG Han-Wen<sup>1,2</sup>, MENG Xu-Ying<sup>1,2</sup>, ZHOU Ji-Hua<sup>3</sup>, ZHANG Yu-Jun<sup>1,2</sup>

<sup>1</sup>(Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China)

<sup>2</sup>(University of Chinese Academy of Sciences, Beijing 100190, China)

<sup>3</sup>(Jinmei Communication, Chongqing 400030, China)

**Abstract:** Information-centric networking (ICN) transforms the network communication mode from the current host-oriented mode to an information-oriented one. Ubiquitous in-network caching is one of the significant features of ICN, which can effectively alleviate server pressure, as well as decrease the user access latency by allowing any nodes in network to cache. However, due to the lack of distribution awareness of content popularity, there are still many problems with the state-of-the-art ICN caching schemes, such as low cache utilization and lack of reasonable planning of cache location. This study proposes a cache coordination scheme based on two-level cache (CSTC) to solve these problems. The content store (CS) of each node is divided into two parts: popularity perception and collaboration allocation. Different caching strategies are applied to cached content with different popularity. At the same time, combined

\* 基金项目: 国家自然科学基金(61572474, 61672500, 61902382, 61972381); 国家重点研发项目(2016YFE0121500, 2018YFB1800403); 中国科学院战略性先导科技专项(XDC02030500)

Foundation item: National Natural Science Foundation of China (61572474, 61672500, 61902382, 61972381); National Key Research and Development Program of China (2016YFE0121500, 2018YFB1800403); Strategic Priority Research Program of Chinese Academy of Sciences (XDC02030500)

收稿时间: 2018-12-25; 修改时间: 2019-06-20; 采用时间: 2019-11-07

with the popularity filtering and routing mechanism, this scheme reduces cache redundancy and optimizes cache location. Finally, simulation experiments based on real network topology show that CSTC has doubled the number of cached secondary popular content. The cache hit ratio has increased by nearly 50%, and the average round-trip hop count is superior to the existing on-path caching method in most cases.

**Key words:** information-centric networking; content-centric network; in-network caching; caching; collaborative caching

现行的以 TCP/IP 为基础的互联网体系结构设计之初是为了解决计算机间点对点的通信需求,然而随着互联网规模和业务类型的爆炸式增长,互联网出现了许多新的问题,内容的高效传输、移动性和安全性等问题亟待解决<sup>[1]</sup>.互联网的主流应用模式逐渐转变为以视频分发、文件下载为代表的信息获取服务<sup>[2]</sup>.为了有效解决以 IP 为中心的网络体系架构在新时代下所存在的各种问题,近年来,一些以信息为中心的互联网体系架构被提出,这种架构即为信息中心网络(information-centric networking,简称 ICN)<sup>[3]</sup>.

ICN 的本质是要将网络通信模式从当前的以位置为中心转变为以信息为中心,即实现位置(服务器/主机的 IP 地址)到内容(用户/应用关心的信息)的转变.这就要求 ICN 中的节点拥有一定的缓存能力,使得用户需要获取信息时,不再向信息源所在的主机地址进行请求,而是直接基于内容标识向网络发起请求.在 ICN 中,缓存呈现泛在化的新特性,网络内任意节点具有缓存能力,以此来缓和服务器的压力,减少网络中的流量,降低用户的访问延迟.

缓存策略是实现 ICN 网络潜在优势的关键技术,得到了学术界的广泛关注.然而,针对如何有效利用缓存资源、提升网络性能的问题,目前并未达成广泛共识<sup>[4]</sup>.Psaras<sup>[5]</sup>和 Dabirmoghaddam<sup>[6]</sup>等人认为:应尽量将内容缓存在网络边缘位置,以靠近用户降低访问延迟.Rossi 等人<sup>[7]</sup>则认为:只有将内容缓存在网络中心,才能保证单位缓存的复用效率,提高网络缓存的利用率.Wang<sup>[4]</sup>,Li<sup>[8]</sup>等人主张通过显示协作方式增加缓存多样性,提升缓存整体效用.而 Zhang<sup>[2]</sup>,Gill<sup>[9]</sup>等人则认为:显示协作方式需要频繁地交换信息与计算开销,这将给以线性速度为要求的高速信息中心网络带来了新的性能瓶颈.当前,对 ICN 网络化缓存的研究还处于起步阶段,现有各种方案大都只侧重某单一方面的性能提升.各缓存策略在缓存内容的多样性和可用性等方面仍有很大的提升空间.

许多研究表明:网络中内容的热度服从 Zipf 分布<sup>[10]</sup>,多数请求往往只集中在少数热门内容上,请求最多的内容称为最热门内容,其次是次热门内容,剩余长尾热度的内容为非热门内容.通过前期大量的仿真实验,我们发现:现有各种方案在达到稳态时,缓存空间仍被大量非热门内容占据,已缓存的内容往往是最热门内容,次热门内容难以稳定缓存,缓存的利用率仍比较低.为提高缓存的利用率,我们希望能够尽可能合理利用缓存空间,稳定缓存更多次热门内容;同时,为了降低延迟,最热门内容也应适度冗余.通过对仿真实验结果的比对与分析,我们发现:理想状态下,最热门内容应适度冗余在网络边缘节点,次热门内容稳定缓存并呈现多样性.

针对现有 ICN 网络化缓存利用率较低和缓存位置缺乏合理规划的问题,本文提出了一种基于两级缓存的协同缓存机制(a cache coordination scheme based on two-level cache,简称 CSTC).主要贡献包括:

- 1) 提出一种分级缓存框架,将每个节点的缓存空间分为 RawCache 和 HashCache 两部分:针对 RawCache,各节点基于本地热度感知进行独立缓存决策;针对 HashCache,通过哈希机制构成域内多节点间协作,实现了域内热度的分布感知及决策,为不同层次的热度内容提供不同的缓存策略;
- 2) 在此框架下,提出了热度筛选机制与路由策略,以实现缓存协作.基于内容热度优化了缓存位置,降低了缓存的冗余,增大了域内稳定缓存内容的数量,从而提高缓存命中率,降低用户请求响应时间;
- 3) 通过仿真实验与现有 5 种主要的 ICN 缓存策略进行了性能比较,并进一步分析了内容热度分布、缓存大小等因素对各缓存策略性能的影响.实验结果表明:CSTC 将次热门内容缓存数量提升了 2 倍;缓存内容数量的增加及缓存位置的优化使得命中率大幅提升,在缓存空间有限的情况下,即使与以高命中率为优势的哈希缓存策略相比,CSTC 最高可将命中率提升 45.4%;同时,CSTC 有效降低了用户请求的响应时间,多数情况下,平均请求响应往返跳数优于现有主要的 5 种缓存策略.

本文第 1 节对相关研究工作进行介绍,并分析了现有 ICN 缓存策略的问题所在.第 2 节详细介绍基于两级缓存的协同缓存机制的运行过程.第 3 节通过仿真实验,从缓存分布、请求命中率等方面对方案进行定量评价.

最后总结全文.

## 1 相关工作及问题分析

缓存策略作为 ICN 的重点研究领域,受到学术界的广泛关注.近年来,各种各样的缓存策略被相继提出.根据不同的分类侧重点,可以将缓存策略分为不同的种类<sup>[11]</sup>.例如:根据内容缓存位置,可以将其分为 On-path 缓存策略和 Off-path 缓存策略;根据各节点协作关系,可以将其分为显式协同缓存策略和隐式协同缓存策略;根据网络节点是否都具备缓存能力,可以将其分为同构缓存策略和异构缓存策略.本节主要就 On-path 缓存策略、基于哈希的缓存策略和混合缓存策略展开讨论.

### 1.1 On-path 缓存策略

在 On-path 缓存策略中,请求内容沿返回路径缓存,而不会缓存到这条路径之外的其他节点.这类缓存策略简单易实现,也在实际部署中被广泛采用.LCE(leave copy everywhere)<sup>[12]</sup>缓存策略是一种典型的 On-path 缓存策略,在该策略中,各节点无差别地缓存途经的所有内容.LCE 虽然易于实现,却不可避免地造成缓存内容的大量冗余,缓存空间的浪费.

Prob(copy with probability)和 ProbCache<sup>[5]</sup>提出了路径上的节点按概率缓存内容的方法,优化 On-path 缓存策略.前者指每个沿途节点都以某一固定概率缓存内容:一方面,每个节点虽然以相同概率对内容进行缓存,但其随机缓存的内容却不一样,提高了缓存的多样性;另一方面,某一内容越热门,则其被缓存的概率越大.后者是 Psaras 等人提出的一种基于加权概率的缓存策略,在这种策略下,越靠近请求者的节点、缓存空间越大的节点,缓存该内容的概率越大.该方法不仅降低了缓存的冗余,还提升了共享路径节点的效用.

为了有效降低缓存冗余,提高缓存空间利用率,有人<sup>[7,9,13,14]</sup>开始倡导将内容缓存在请求路径上权重最高的一个节点上来降低缓存冗余,他们对于节点权重的定义各不相同.Rossi 等人<sup>[7]</sup>研究了各种基于网络拓扑中心度的缓存策略,并得出,将内容缓存至中心度数最大的节点处缓存效果相对较好.Xu 等人<sup>[13]</sup>设计了基于动态 LRU 队列和基于布隆过滤器两种内容效用统计方法,将内容缓存在路径上效用最大的节点上.Ren 等人<sup>[14]</sup>提出的 Magic 缓存策略将内容缓存在收益最大的节点上,并定义了在不同节点的缓存收益计算公式,缓存收益由内容在该节点的局部热度和节点到请求者的距离两方面共同决定.然而,文献并没有明确给出内容热度统计方式.Gill 等人<sup>[9]</sup>提出了 BidCache 竞拍缓存模型,请求转发过程中,各节点根据当前节点信息进行“出价”,请求包头记录了最大“价格”及其出价节点.当内容返回时,将其缓存至路径上出价最高的节点上.

各种 On-path 缓存策略的共同特点是,网络节点只与其请求路径上游或下游的某些节点进行协作.这一共性限制了网络化缓存的潜在性能提升.由于缺乏全局拓扑视图与缓存状态相关信息,缓存策略只能基于单节点、单条传输路径对缓存进行优化,这就限制了网络性能提升的上界,也带来了如下几个方面的问题.

- 1) 缓存大量冗余.由于各节点独立进行缓存决策,同一网络中各节点请求分布又基本一致,导致同一层各节点缓存下的内容基本相同.有限的缓存空间和大量重复的内容,给 ICN 网络带来了更大的挑战;
- 2) 命中退化现象.热门内容被拉取到网络边缘后,多数请求在边缘节点得到满足.边缘节点的过滤效应,使得靠近中心的节点收到的内容请求分布趋于随机,加上下游节点的请求汇聚作用,中心节点收到的请求热度分布相对混乱.这种过滤与汇聚效应,使得原本在下游节点没有命中的请求,在上游节点命中的概率也很低;
- 3) 缓存可用性差.缓存仅为路径可见,即使缓存在请求节点的邻居节点,若不在其请求路径上,则缓存不可用.缓存的可用性对网络性能有很大的影响,较高的可用性,能够提升单位缓存空间的效用.

### 1.2 基于哈希的缓存策略

为了有效解决 On-path 缓存机制中缓存大量冗余、缓存命中率低的问题,曾在 P2P 中广泛应用的基于哈希的缓存策略得到了人们的重新关注.这类策略将内容的放置与路由请求通过哈希机制相结合,哈希映射不仅决定了内容应该缓存到哪,也决定了请求应该向哪转发.通过这种请求的重定向机制,大幅降低了缓存的冗余,也

提高了缓存的命中率,避免了频繁的信息开销。

Thar 等人<sup>[15]</sup>给出了一种基于哈希的缓存模型,阐述了包括协作转发、热度预测、内容存取等的详细过程。仿真实验显示:该方案在缓存命中率方面有着巨大提升,然而也导致了内容传输跳数上的增加。

传输跳数增加是所有哈希缓存策略存在的一个共同问题,下面分别是不同的解决方案<sup>[16-19]</sup>。Saino 等人<sup>[16]</sup>提出了两种不同的解决方法:一种使内容数据返回时,总是沿着最短路径返回;另一种则通过广播的方式,多路径返回内容数据。Sourlas 等人<sup>[17]</sup>认为:缓存命中率与传输跳数相互制约,当协作域较小时,虽然缓存的整体命中率有所下降,但哈希策略引发的跳数增加也被限定在了一定的范围之内。作者主张将大的协作域划分为多个小的协作域,并给出了两种协作域划分算法。Wang 等人<sup>[18]</sup>设计了一种缓存分配算法,该算法在限定的跳数增加范围内,尝试寻找一种缓存分配方案。内容首先哈希映射到一个 Partition,每一个 Partition 对应一个或多个缓存节点。由于内容分配方案由入口路由器请求分布、网络拓扑、内容热度等多种因素共同决定,当网络状态发生改变时,尤其是不同入口请求流量发生改变时,若不改变已有分配方案,网络性能则会明显下降。而改变内容分配方案带来的代价是极大的,需要改变不同节点的缓存大小、路由转发表、Partition 映射表等等。Li 等人<sup>[19]</sup>提出了一种启发式内容分配方案,将协作域内的节点划分为与出口路由器数量相等的簇。该算法采用模拟退火的方式,每次交换不同簇种的两个节点,看是否优于先前分配方案。

基于哈希的缓存策略通过设计一种内容分配方案,将内容的缓存任务分配到域内特定的网络节点。这种方式有效解决了缓存冗余问题,提高了缓存命中率,但也带来了一些新的问题。

- 1) 缓存位置缺乏合理规划:内容缓存位置被预先决定、将指定内容与特定节点进行绑定的缓存方式,一定程度上违反了 ICN 内容与位置分离的设计原则。现有的各种哈希缓存策略内容分配机制缺少对热度内容缓存位置的合理规划。当内容热度随着时间、地域发生变化时,这种问题尤为明显;
- 2) 传输路径的增加:相比 On-path 缓存策略中最短路径路由策略,基于哈希的缓存策略需要先将请求转发至特定节点,若未命中,再转发至内容源节点。这种路由策略增加了内容请求与回传中的途经跳数。这也是现阶段各种基于哈希缓存策略着重关注的问题。目前,多数方案认为影响传输跳数的因素与网络拓扑、请求分布密切相关,忽略了内容热度对传输跳数的影响。

### 1.3 混合缓存策略

关于结合不同缓存策略的缓存机制最近已有不少研究,主要包括基于网络结构的混合缓存策略以及基于内容热度的混合缓存策略。

基于网络结构的混合缓存策略根据节点所处位置为不同节点提供不同的缓存策略。Zhang 等人<sup>[20]</sup>提出将网络分成一个核心域和多个边缘域,在核心域使用基于哈希的缓存策略,在边缘域使用 On-path 的缓存策略。该方法实际上是一种多域协作的部署策略。节点间缓存策略的融合,虽然在整体上提升了网络性能,但对于单个域本身来说,缓存性能并没有改变,因为域内应用的还是原有的缓存策略。该方法并不能解决缓存内容在边缘节点过度冗余的问题。

基于内容热度的混合缓存策略为不同热度的内容提供不同的缓存策略。近年来,已有许多缓存策略都考虑了内容热度对缓存的影响:Yu 等人<sup>[21]</sup>提出了基于热度的动态缓存权限策略;Yovita 等人<sup>[22]</sup>考虑将内容按照不同的服务分成不同类;Yu 等人<sup>[23]</sup>将路由器分级,将热度高的内容存储在等级高的路由器上。相应的混合策略也得到了人们的广泛关注,Li<sup>[24]</sup>和 Chang<sup>[25]</sup>等人通过理论建模分析,证明了基于热度合理配置缓存空间,使其一部分缓存最热门的内容,另一部分参与全局协作,缓存次热门内容,能够有效提高网络性能。这种基于单个节点缓存空间划分,有效利用内容热度分布的机制,使两种缓存策略相结合,达到了优势互补的效果。文章虽然强调了内容热度的重要性,却没有给出内容热度的统计与筛选机制,且基于信息交换的域内协作机制也带来了频繁的计算开销与信道占用。总的来说,这类方案目前主要集中在理论建模与数学分析,并没有给出可供实际部署的缓存策略。

本文提出的 CSTC 是一种基于两级缓存的协同缓存机制,不同于 Li<sup>[24]</sup>和 Chang<sup>[25]</sup>等人通过域内信息交换来协作缓存,我们引入哈希机制,避免了大量的信息传递与计算开销。同时,我们也给出了两部分缓存的协作方

式、替换策略、热度筛选方法及路由策略,是一种可应用于实际部署环境的缓存机制。

## 2 CSTC 运行机制

CSTC 通过引入两级缓存机制,实现了不同热度内容的分级缓存,一方面降低了缓存的冗余;一方面,利用内容热度对缓存位置进行了合理的规划,使得缓存分布更接近理想分布状态。每个节点的缓存空间被分为 RawCache 和 HashCache 两部分:针对 RawCache,各节点基于本地热度感知进行独立缓存决策;针对 HashCache,则通过哈希机制构成域内多节点间协作,实现了域内热度的分布感知及决策,为不同层次的热度内容提供不同的缓存策略。通过所设计的热度筛选机制与路由策略,达到了内容热度筛选的目的,优化了缓存位置,降低了缓存的冗余,增大了域内稳定缓存内容的数量。本节将阐述 CSTC 的详细过程。

### 2.1 CSTC基本框架

研究表明:网络中内容的热度服从 Zipf 分布<sup>[10]</sup>,不同内容的请求频率相差巨大,多数请求仅仅集中在少数热门内容上。Li 等人<sup>[24]</sup>通过构建模型,分析如何配置单个路由节点的缓存空间,以达到整体网络性能与成本的最优;Chang 等人<sup>[25]</sup>在 Li 等人研究的基础上提出一种确定节点缓存空间比例以及如何放置内容的方法,证明将缓存空间基于内容热度分块——一部分采用 On-path 来缓存最热门的内容,另一部分参与全局协作,缓存次热门内容,可以在保持缓存命中率的同时,有效降低请求的平均时延。

然而,上述方法仅仅是对最优缓存策略的数学量化与理论建模,缓存策略也是基于所有内容热度已知的先决条件下完成的,同时,各节点的缓存依赖于集中式的决策控制,并不适用于实际环境的部署要求。CSTC 基于以下考虑。

- 1) 最热门内容应在网络边缘适度冗余,减少用户请求跳数。Fayazbakhsh 等人<sup>[26]</sup>通过大量仿真实验证明:仅仅简单地在网络边缘部署足够大的缓存,就能达到可观的网络性能提升。这正是热门内容在网络边缘冗余带来的效果。让用户的多数请求在网络边缘节点得到满足,不仅减少了用户的请求时间,也极大降低了网络负载;
- 2) 次热门内容应稳定缓存,尽量避免非热门内容占据缓存空间。最热门内容只占网络中内容的极少部分,网络中存在着大量的次热门和非热门内容。为有效提高请求命中率,我们可以选定合理的存储位置,通过请求重定向汇聚次热门内容的热度,使得次热门内容能在指定节点稳定缓存,进而提高次热门内容缓存多样性,降低非热门内容对缓存空间的占用;
- 3) 节点间较小的信息交换开销与线性算法复杂度。现有的各种全局节点协作的内容热度感知方案<sup>[8,27]</sup>需要不同节点间频繁的信息交流与计算开销,同时会给集中控制节点带来较大的流量负担与计算压力,这会给以线性速度为要求的高速信息中心网络带来新的性能瓶颈。

CSTC 中,节点缓存空间被分为 RawCache 和 HashCache 两部分,如图 1 所示。RawCache 上采用 On-path 缓存策略,对途经的内容进行缓存,同时对缓存下来的内容进行基于单节点热度统计。通过这种方式,最热门内容被拉取到网络边缘稳定缓存,而次热门与非热门内容则会在 RawCache 中频繁替换,难以稳定缓存。当 RawCache 中缓存内容发生替换时,若为非热门内容,则直接替换出缓存空间,否则根据哈希映射找到其匹配节点,并将其发往对应节点的 HashCache。不同节点的 HashCache 构成一个协作整体,一方面,每块 HashCache 通过哈希机制缓存不同的内容,有效避免了缓存冗余;另一方面,域内被替换出的不同内容会被汇聚到不同节点,内容会依照其全局热度在该节点展开缓存空间的二次竞争。通过这种两级缓存机制,最热门的内容会留在 RawCache 中,次热门内容会稳定缓存在 HashCache 中。不同节点的 HashCache 被分配以不同的缓存任务,这些节点的 HashCache 通过哈希机制构成一个协作整体。

节点上相互关联两部分的缓存可以采用 LRU 或 LFU 缓存替换策略。初始阶段,两部分缓存都为空,当内容经过节点时,节点的 RawCache 部分执行 On-path 缓存策略对内容进行存储。当 RawCache 剩余缓存空间大小为 0 时,若仍需缓存新内容,则会通过缓存替换决策替换掉原有的缓存内容。这时,节点先依照所统计的内容命中次数判断将被替换出的内容的热度:若为热门内容,则查找哈希映射表,将内容发往对应节点的 HashCache,再执行

替换操作;否则,直接执行替换操作.协作域中节点的 HashCache 部分收到从 RawCache 发来的内容时,对其进行缓存决策,此时若该节点的 RawCache 中已缓存该内容,则将内容从 RawCache 中删除以避免冗余.当 HashCache 发生内容替换时,考虑到 HashCache 中多为热门内容,节点将替换内容移动到本节点的 RawCache 中,而不是直接丢弃.

通过这种两级缓存机制,域内的 RawCache 会对网络中的内容进行筛选,被频繁访问的最热门内容会稳定地缓存在各个节点的 RawCache 中,降低用户的访问跳数,也减轻了网络核心节点的负载.RawCache 中替换出的仍有一定热度的内容会通过哈希映射缓存在 HashCache 中,大幅提高了缓存的多样性,同时避免了过度冗余.

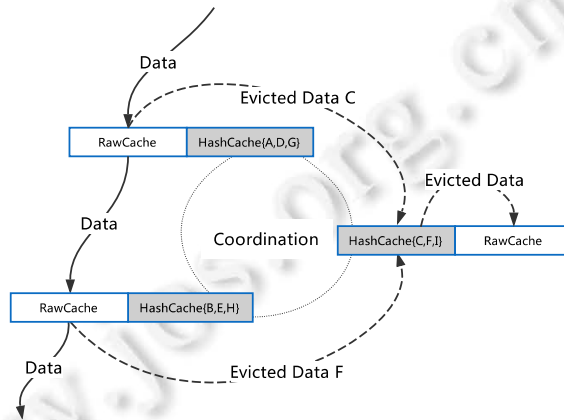


Fig.1 Two-level cache for CSTC

图 1 CSTC 两级缓存

## 2.2 CSTC 热度筛选

内容热度是影响网络缓存性能的关键.相比于常见的利用内容请求次数来表示内容热度,本节设计了基于内容命中次数的热度统计方法,主要基于以下考虑.

- 1) 由于网络中的内容非常多,统计所有内容的请求数是不现实且不必要的.为了降低时间空间开销,我们只针对已缓存的内容进行请求次数的统计,已缓存内容的请求次数即为内容的命中数;
- 2) 在 CSTC 方法中,热度内容首先基于 On-path 策略存入 RawCache,针对 RawCache 中已缓存的内容,再利用其热度信息判断是否存入 HashCache.热度统计只是针对 RawCache 已缓存内容,命中数可以很好地反映已缓存内容的热度.

为了准确感知内容热度,CSTC 中每个节点需要维护两种数据结构:本地命中列表(local hit table,简称 LHT)和热度统计列表(popularity table,简称 PT).如图 2 所示,LHT 用以维护节点已缓存内容在该节点上的命中次数,格式为 $\langle name:(hit,time) \rangle$ .PT 用以维护内容在协作域内的热度,格式为 $\langle name:(popularity,time) \rangle$ .各字段含义如下.

- *name*:目标内容名字;
- *hit*:内容命中次数,非负整数;
- *popularity*:内容在域内的热度,非负整数;
- *time*:列表项上次更新时间.

当内容在节点的缓存空间(RawCache 或 HashCache)命中时,LHT 对应位置上的 *hit* 项则加 1.LHT 的列表长度与节点的缓存大小一样,当内容从节点缓存空间移除时,对应表项也随之移除.因此,LHT 统计的内容命中次数只是单节点局部时间热度,保证了列表较小的空间复杂度.*hit* 的初始值为对应内容在 PIT 表中记录的返回接口数目减 1,表示在内容被缓存到节点的 CS 中之前,内容的热度等于内容的请求数.当内容被缓存在节点的 CS 中后,内容在该节点的请求可以在 CS 中命中,此时内容的请求数等于内容的命中数.当 Interest 在节点的 CS 中命中时,使用下面的公式更新表项:



$$x.hit = x.hit \times \rho^{t_{cur} - t_{last}} + 1 \tag{1}$$

其中,  $\rho(0 < \rho < 1)$  决定了内容命中次数随时间衰减的快慢,  $t_{cur}$  指当前时间,  $t_{last}$  指上次表项更新时间. 与周期性进行热度衰减的方法不同, 我们采用基于命中的时间衰减方法, 这种方法不需要周期性地更新内容热度, 只需要在有数据包到来的时候进行衰减操作, 降低了更新的开销. 当数据包到达时, 首先利用时间衰变参数对历史的命中数进行衰减, 再将衰减操作后的值加 1, 表示当前内容在节点中的热度, 当前时间与上次表项更新时间间隔越久, 历史命中数对当前内容热度的影响越小.

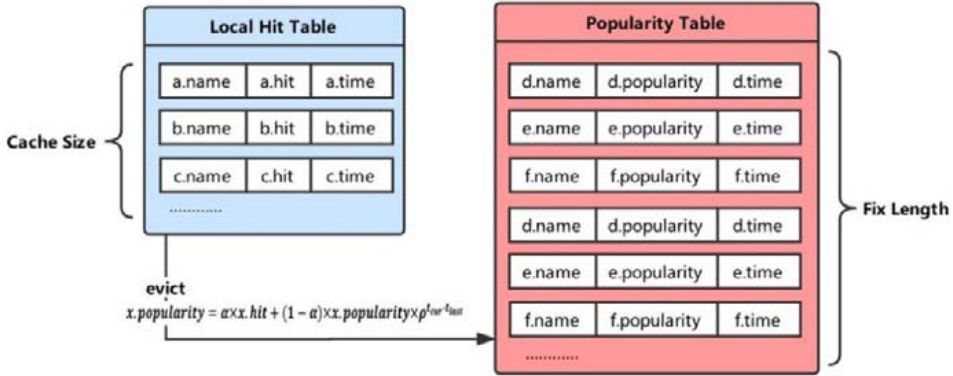


Fig.2 Local hit table and popularity table

图 2 本地命中列表与热度统计表

当缓存内容在 RawCache 中发生替换时, 首先根据 LHT 表中 hit 项判断其命中次数是否大于所设阈值, 若满足要求, 则认为该内容应该被缓存至域内 HashCache 中, 此时节点会将内容本身以及该内容在 LHT 表中的相关信息一并发送至哈希映射所得节点. 对应节点收到内容以后, 除了存储内容本身以外, 还会更新自身 PT 表项. 更新规则如下: 若 PT 表中没有该项内容, 则直接将内容的单节点命中次数作为内容热度, 即  $x.popularity = x.hit$ ; 若 PT 表中已存在该项内容, 则使用下面的公式更新表项:

$$x.popularity = \theta \times x.hit + (1 - \theta) \times x.popularity \times \rho^{t_{cur} - t_{last}} \tag{2}$$

其中,  $\theta(0 < \theta < 1)$  是单节点内容命中次数在域内内容热度上的权重参数,  $\rho(0 < \rho < 1)$  决定了内容热度随时间衰减的快慢,  $t_{cur}$  指当前时间,  $t_{last}$  指上次表项更新时间. 为与 LHT 表的时间衰减保持一致, 我们在 PT 表的热度更新中同样引入时间衰减. 通过哈希映射机制, 域内各节点 RawCache 中替换出的同一内容会被发送到同一节点, 目标节点通过域内各个节点的局部命中次数不断更新该内容的 popularity, 最终得到的 popularity 实际上是域内所有节点命中次数共同作用的整体结果. PT 表中内容按照 popularity 由大到小排序, 用以指导 HashCache 缓存哪些内容. 节点会记录 HashCache 中已缓存内容的 popularity 最小值, 当要缓存新内容时, 首先比较二者的 popularity: 若要缓存的内容的 popularity 大于该值, 则进行缓存, 并替换掉 popularity 最小的内容; 否则不做缓存处理.

### 2.3 CSTC路由策略

本节假设我们提出的 CSTC 缓存机制运行在典型的 ICN 网络 NDN<sup>[12]</sup> 中, NDN 中共有两种数据类型: 兴趣分组 (interest packet) 和数据分组 (data packet). 用户发起的请求称为 Interest, 响应请求返回的内容数据称为 Data. 需要说明的是: HashCache 上采用的哈希映射机制的详细过程并不是 CSTC 关注的主要内容, 就像我们并不限制 RawCache 上必须采用某种 On-path 缓存策略一样.

表 1 中, 当节点收到 Interest 请求时, 首先会查找自身内容存储表 (content store, CS) 与待定请求表 (pending interest table, 简称 PIT), 看是否有内容命中. 若都没有命中, 则根据哈希映射得到目标节点, 并向其转发 (第 5 行). 若 Interest 到达目标节点后仍没有命中, 则将 Interest 转发至内容源 (第 3 行).

Table 1 INTEREST routing algorithm

表 1 INTEREST 路由算法

---

算法 1. 收到 INTEREST 包.

---

```

1: if 缓存未命中 and 未匹配到 PIT 表项 then
2:   if 到达目标节点 then
3:     转发至内容源
4:   else
5:     转发至哈希映射所得目标节点
6:   end if
7: else
8:   返回 DATA 包
9: end if

```

---

表 2 中,节点收到 Data 数据分为两类:一类是普通的 Data,另一类是其他节点 RawCache 中替换出的内容重新组装成的 Data.若是普通 Data(第 14 行),则按照 RawCache 上采用的 On-path 缓存策略对其进行处理,同时,如果在缓存的过程中发生了内容替换,则节点将内容本身及其所对应的 PT 表项封装为新的 Data,转发至哈希映射所得目标节点,转发过程中的途经节点不再对其进行处理(第 12 行).当目标节点收到其他节点发送来的 Data 时(第 2 行),则将其缓存至 HashCache 中并按照公式(2)更新 PT 表项.HashCache 中替换出的内容会被移动至该节点的 RawCache 中,而不会直接丢弃.

Table 2 DATA routing algorithm

表 2 DATA 路由算法

---

算法 2. 收到 DATA 包.

---

```

1: if Data 是 RawCache 中替换出的内容 then
2:   if 到达目标节点 and Data.popularity>min(PT.popularity) then
3:     if 该节点 RawCache 已有该内容 then
4:       移动至 HashCache and 更新 PT.popularity
5:     else
6:       缓存至 HashCache and 更新 PT.popularity
7:     end if
8:   if 替换掉了其他内容 then
9:     将替换出的内容移动至该节点 RawCache
10:  end if
11: else
12:  转发至下一跳
13: end if
14: else if 是普通 DATA 包 and 匹配到 PIT 表项 then
15:  执行 On-path 缓存策略处理该内容
16:  if 替换掉了内容 and hir>阈值 then
17:    将替换掉的内容连同所对应 PT 表项组装为 DATA 包
18:    直接转发至哈希映射所得目标节点
19:  end if
20:  转发普通 DATA 包至下一跳
21: end if

```

---

### 3 仿真实验

为了评估 CSTC 的缓存效果与性能,我们基于 ccnSim<sup>[28]</sup>设计实现了缓存仿真实验,将 CSTC 与多种现有缓存方案进行了对比分析实验,并探究了各缓存因素对缓存性能的影响.本节将从实验环境与配置、对比方案与性能指标、仿真结果与分析这 3 个方面详细阐述仿真实验与结果.

#### 3.1 实验环境与配置

考虑到一个请求域的大小,我们选择了美国爱荷华州的一个真实拓扑结构<sup>[29]</sup>作为仿真拓扑网络,该拓扑共有 30 个节点、38 条边.为了合理模拟真实网络状态,当内容在域内未命中时,我们设置的出域路由的请求跳数为 10 跳.每个节点都连接有一组用户,不间断地连续模拟发起内容请求,用户请求服从泊松分布.仿真实验中共有 100 万个不同的内容,我们假设内容整体上服从参数为  $\alpha$  的 Zipf 分布,也就是说,排名第  $k$  的热门内容的请求



概率为:

$$p_k = \frac{1}{\sum_{i=1}^N \left(\frac{1}{i^\alpha}\right)} \quad (3)$$

实验中,缓存默认替换算法采用 LRU(least recently used),每次替换掉最近最少使用的内容.仿真过程中,首先等待每个节点的缓存空间被占满,然后等待每个节点处于稳定状态(即节点收到的请求数上下波动处于较小范围)后,开始数据收集与统计.仿真实验中,各项参数见表3.下文未特别说明实验参数的情形下,均采用表3中默认值.

Table 3 Parameters of experiments

表3 实验参数

参数	默认值	变动范围
Zipf 参数 $\alpha$	0.8	0.6~1.4
缓存大小	500	500~7 000
内容源距离域跳数	10	
内容数量	1 000 000	
缓存替换算法	LRU	
时间权重参数 $\rho$	0.9	

### 3.2 对比方案与性能指标

仿真实验中采用的缓存方案包括如下几种.

- CSTC:即本文提出的基于两级缓存的缓存机制.实验中,每个缓存节点的 RawCache 与 HashCache 各占一半,RawCache 上采用  $Prob(0.05)$  的 On-path 策略,设置的命中阈值为 1,热度权重参数 $\theta$ 为 0.6,时间权重参数 $\rho$ 为 0.9;
- OH:我们实现了文献[15]中的哈希映射机制,将其应用到仿真实验中,以代表现有各种基于哈希的缓存策略,我们将这类策略简称为 OH(only Hash);
- LCE:多数 ICN 默认缓存策略,即节点无差别的缓存所有内容<sup>[12]</sup>;
- $Prob(p)$ :每个节点以固定概率  $p$  缓存经过的所有内容,我们设置的  $p=0.05$ ;
- ProbCache:一种基于加权概率的缓存策略,距离请求者越近、存储空间越大的节点,缓存该内容的概率越大;
- BidCache:我们实现了文献[9]中的 BidCache 缓存机制,这是一种近年来较为典型的 On-path 协作式缓存策略,它利用较少的代价进行路径上节点的通信,结合网络的拓扑实现缓存决策.

本文主要从两个方面评判缓存方案的优劣.

#### (1) 请求命中率

内容请求在域内总的命中率反映了域内整体缓存状态(包括缓存多样性、缓存可用性、缓存冗余度等)的好坏,命中率越高,说明缓存效果越好,带来的网络性能提升也越明显.请求命中率的计算公式如下:

$$HitRate = \frac{\sum_{i=1}^{|V|} Num_{hit}}{Num_{total}} \quad (4)$$

其中, $Num_{hit}$  表示请求在该节点的命中次数, $Num_{total}$  表示请求总数, $i$  表示第  $i$  个节点.

#### (2) 平均传输跳数

平均传输跳数不仅反映了缓存内容的好坏,还体现了缓存位置是否合理.当请求命中率较高时,多数请求在域内命中,避免了出域的跳数代价,跳数也就越小.同时,缓存内容位置越合理,平均传输跳数也就越小,用户访问延时越低.其计算公式如下:

$$AverageHops = \frac{\sum hop_{interest} + hop_{data}}{2 | Num_{total} |} \quad (5)$$

其中,  $hop_{interest}$  表示 Interest 请求所用跳数,  $hop_{data}$  表示 Data 返回所用跳数,  $Num_{total}$  指总请求数.

### 3.3 实验结果

#### 3.3.1 缓存内容分布

前文分析已知:在缓存空间一定的情况下,能够在本地域内缓存的内容数量越多,缓存内容的热度越高,越能高效响应用户的不同请求.本节分析了不同缓存策略下缓存内容的分布状态,内容总数为 1 000 000.根据 Zipf 分布定律,我们将热度排名为 1~1 000 的内容用黑色表示,代表最热门内容;排名在 1 000~10 000 的内容用灰色表示,代表次热门内容;其他内容用白色表示,代表非热门内容.图 3 在真实拓扑上绘制了各个节点在不同缓存方案下缓存内容分布状况.

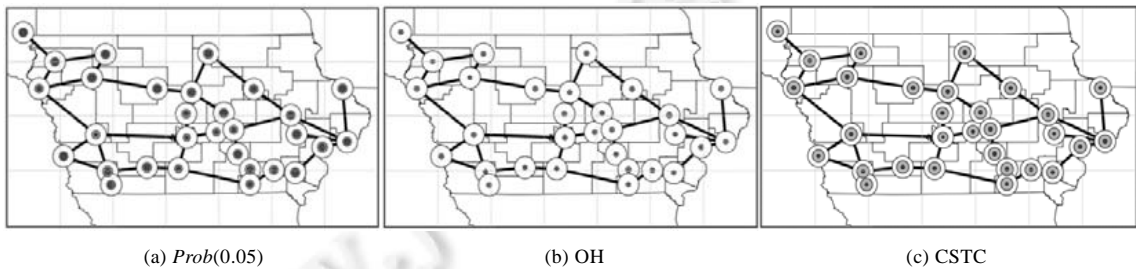


Fig.3 Distribution of cache

图 3 缓存内容分布

从图 3(a)可以看出:Prob(0.05)策略下,各节点缺乏合理协作机制,最热门的内容在各节点上形成了大量冗余,缓存空间被最热门的一小部分内容大量占据,次热门缓存数量极少,缓存多样性较差,缓存空间利用率较低.这也印证了前文所述的 On-path 所存在的各种问题.图 3(b)展示了 OH 策略下各节点缓存状态,与 Prob(0.05)不同的是:在该策略下,虽然有效避免了缓存冗余,缓存的多样性得到了极大的提升,但缓存的内容大多为非热门内容,缓存内容的效用较差.

与前两种缓存机制相比,CSTC 缓存效果最为理想.如图 3(c)所示,各个节点对最热门内容产生适度冗余,在次热门内容上缓存种类较多,有效避免了非热门内容对缓存空间的过量占据,这主要是因为 RawCache 的内容热度感知保证了 HashCache 中缓存内容的热度,通过基于哈希的请求重定向,使得这些次热门内容的请求被汇聚并稳定缓存到特定节点.

#### 3.3.2 缓存大小的影响

缓存大小决定了缓存内容的多少,是影响缓存性能的重要因素之一.本节分析了不同缓存空间大小对缓存性能的影响.我们设置的缓存空间大小变动范围为 500~7000,其他实验参数保持不变,观察请求命中率与平均传输跳数的变化.

从图 4(a)可以看出,请求命中率随着缓存空间的增大而增大.CSTC 与 OH 两种策略在命中率上要显著高于另外 4 种基于 On-path 的缓存策略,这得益于哈希映射机制有效降低了缓存冗余,提高了本地域内的缓存多样性,从而使得更多的用户请求能在本地域内得到响应.从图中可以发现:CSTC 即使内容多样性不如 OH,但命中率却大于 OH.这主要是因为 CSTC 基于热度感知使得缓存下的内容热度较高,能够在适当的位置被稳定缓存;而 OH 虽然可能缓存下更多内容,但由于没有热度感知,使得缓存下的很多是非热度内容,这些内容将在缓存中频繁进出,浪费缓存空间.同时我们发现:当缓存空间较小时,CSTC 的优势更为明显.这是因为缓存空间变大以后,即使 OH 缺乏合理热度感知机制,也能将大多数热门内容缓存下来,从而缩小了与 CSTC 的命中率差距.另外,ProbCache 与 Prob(0.05)两种缓存策略在命中率上基本一致,都高于 LCE, BidCache 的命中率略高于 ProbCache.这是由于 BidCache 不仅考虑与用户的距离,还考虑当前节点的中心度等,只有综合分数最高的节点才能够缓

存内容.因此,BidCache 有更合理的内容放置策略,缓存的内容更多,内容命中率更高.当缓存空间越来越大时,ProbCache 缓存的内容越来越多,两种方法缓存的内容种类差距变小,命中率也就越来越接近.

图 4(b)展示了不同缓存空间大小对平均传输跳数的影响.从图中可以看出:当缓存空间较小时,由于 On-path 缓存策略可以将最为热门的内容缓存在网络边缘,所以在平均跳数上优势较为明显;而 OH 因为需要对所有请求重定向到特定节点,其请求和响应的平均跳数最大.但随着缓存空间的增大,CSTC 与 OH 在跳数上的下降较为明显,而另外 4 种 On-path 策略对缓存空间大小的变化并不敏感.这是因为当缓存空间增大时,CSTC 与 OH 可以缓存下来更多不同的内容,从而大幅减少转发至域外的请求数量,多数请求在域内命中,总的平均跳数也随之下降.

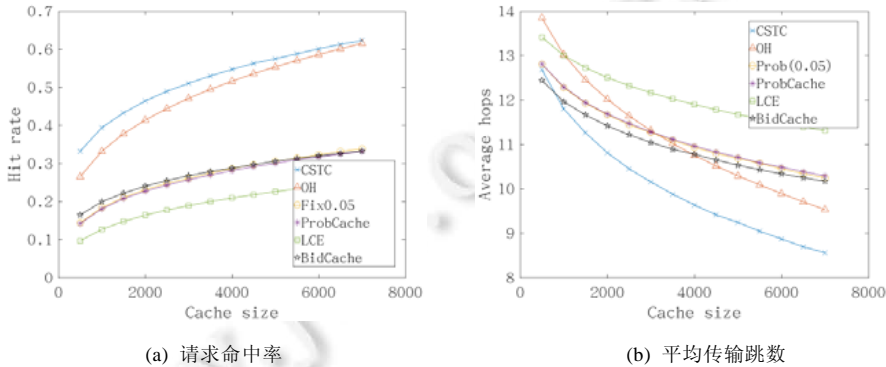


Fig.4 Impact of cache size

图 4 缓存大小的影响

3.3.3 内容热度的影响

已有研究表明,网络中内容热度分布服从 Zipf 分布.本节探究了 Zipf 分布参数 $\alpha$ 对缓存性能的影响.从图 5(a)中我们可以看出:随着参数 $\alpha$ 值的增大,内容的热度收敛更加明显,更多的请求集中在少数的热门内容上,导致请求的整体命中率提高.这是因为各节点只需将较为热门的少许内容进行存储,便能达到较高的命中率.从图中可以看出:CSTC 的命中率要高于 OH,但随着参数 $\alpha$ 的增大,二者的命中率逐渐趋于相同.这是由于参数 $\alpha$ 较小时,内容热度较不明显,OH 虽然缓存的内容多样性更丰富,但由于较差的热度感知能力,多数都为请求率较低的非热门内容.图 5(b)展示了不同参数 $\alpha$ 下,各缓存机制的平均传输跳数.CSTC 的平均传输跳数不仅优于简单哈希机制,在多数情形下也优于各种 On-path 缓存机制.这是因为虽然 On-path 缓存机制极大地减少了命中内容的往返跳数,但由于存在着大量的冗余,其缓存内容的多样性较差,命中率也较低.当内容在域内没有命中时,就要将请求转发至域外,导致跳数的大幅增加.

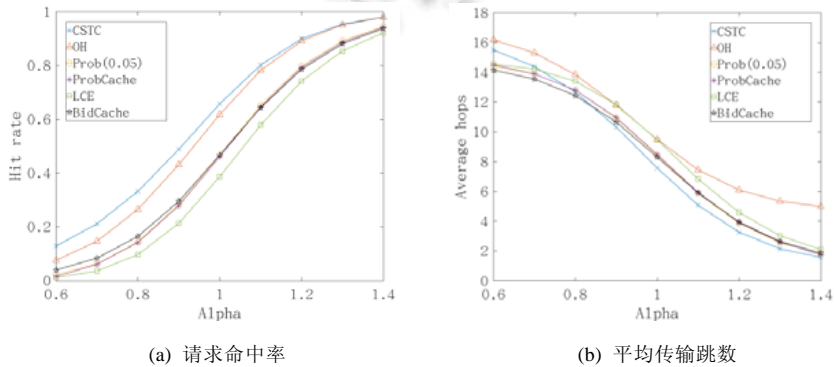


Fig.5 Impact of content popularity

图 5 内容热度的影响

### 3.3.4 两级缓存比例的影响

本节分析了 CSTC 中 RawCache 和 HashCache 两级缓存比例对缓存性能的影响.通过大量实验我们发现:两级缓存比例对网络性能的影响在不同条件下是不一致的,应该结合内容热度分布参数 $\alpha$ 和缓存大小综合考虑.而内容热度分布参数 $\alpha$ 对其影响最为突出,下文以 $\alpha$ 为代表,分析其影响.

图 6(a)描述了两级缓存比例对 CSTC 命中率的影响,横坐标代表 RawCache 所占比例的变化,纵坐标表示请求命中率.图中不同曲线反映了不同 $\alpha$ 下,两级缓存比例对请求命中率的影响.从图中可以看出:随着 RawCache 比例的提高,缓存的命中率在逐渐降低.同时我们发现:当 $\alpha$ 越小,其下降也越明显.当 RawCache 比例提高,也就是 HashCache 比例下降时,域内稳定缓存的次热度内容数量减少,即缓存内容多样性随之下降,从而导致了命中率的降低. $\alpha$ 越小,缓存内容热度分布越不集中,RawCache 基于本地热度感知所能稳定缓存下的热度内容减少,因此命中率会有所下降.从图中可以发现:当 RawCache 比例提高到 100%,即退化为仅基于本地热度感知的一级缓存策略时,命中率下降最为明显.这也说明了两级缓存的设计是缓存性能提升的优势所在.

图 6(b)与图 6(c)展示了两级缓存比例对用户请求响应平均传输跳数的影响.从图 6(b)我们发现:当 $\alpha$ 较小时,随着 RawCache 比例的提高,平均传输跳数在逐渐升高.而图 6(c)显示:当 $\alpha$ 较大时,随着 RawCache 比例的提高,平均传输跳数会先降低再上升.我们分析了其中的原因:当 RawCache 比例提高时,一方面,RawCache 可以通过 On-path 方式缓存更多的内容,On-path 较小的传输跳数会减小 CSTC 的平均传输跳数;另一方面,因 HashCache 比例的下降,会导致许多原本在域内就可以命中的内容,现在需要将其转发至域外,这增加了基于二级缓存的缓存方式的平均传输跳数.所以当 $\alpha$ 较小时,On-path 缓存机制本身命中率较低,其对 CSTC 跳数的降低不足以弥补由于 HashCache 下降所带来的跳数下降,从而展现出整体平均传输跳数的增加;而当 $\alpha$ 较大时,On-path 缓存机制的命中率较高,当其比例提升时,会较大幅度的降低往返跳数,所以 CSTC 的平均传输跳数开始呈现出下降的趋势.但随着 RawCache 的比例不断增大,其带来的跳数下降也越来越微弱,而 HashCache 比例的下降导致的跳数增加开始显现,所以 CSTC 的平均传输跳数开始呈现上升趋势.

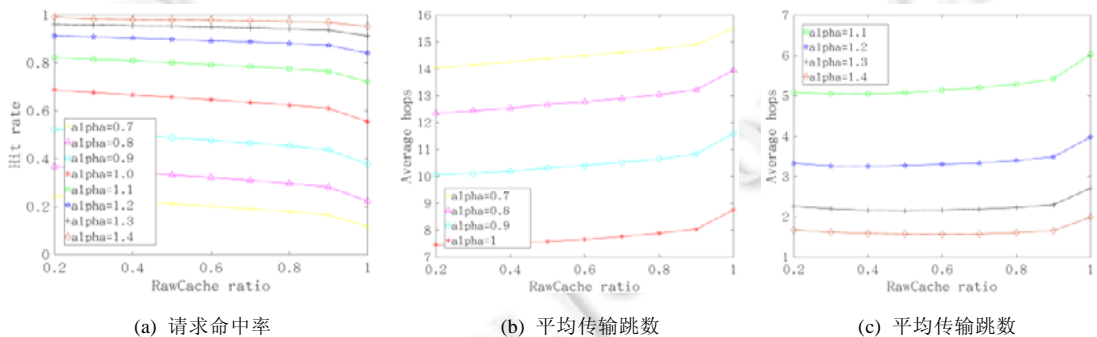


Fig.6 Impact of two-level cache ratio

图 6 两级缓存比例的影响

通过这一小节的分析,我们发现两级缓存比例对请求命中率与平均传输跳数的影响在不同条件下是不一致的,它取决于 On-path 缓存方式的命中率的高低.而影响 On-path 缓存方式的命中率又受内容热度分布参数 $\alpha$ 和缓存大小等其他因素影响,所以我们认为:在考虑两级缓存比例对网络性能的影响前,要先明确其他影响因素,不能一概而论.

## 4 结论

本文针对现有方案存在的缓存内容效用较差、缓存位置不合理的问题,提出了一种基于两级缓存的缓存机制(CSTC).该方案通过引入两级缓存,既保证了缓存的多样性,又合理规划了热门内容的缓存位置.仿真实验表明:CSTC 将次热门内容缓存数量提升了 2 倍;缓存内容数量的增加及缓存位置的优化,使得命中率相比原有方

案有大幅提升,即使与以高命中率为优势的哈希缓存策略相比,在缓存空间有限的情况下,CSTC 最高可将命中率提升 45.4%;同时,CSTC 有效降低了用户请求的响应时间,多数情况下,平均请求响应往返跳数优于现有主要的 5 种缓存策略。内容热度的统计方法是影响 CSTC 的关键因素之一,下一步工作将充分考虑内容请求热度的地域与时域性分布特征,考虑真实网络中内容热度随时间的动态变化,进一步研究内容热度的动态感知方法,以优化 CSTC 的缓存性能。

## References:

- [1] Wang XW, Li J, Tan ZH, *et al.* The state of art and future tendency of “Internet+” oriented network technology. *Journal of Computer Research and Development*, 2016,53(4):729–741 (in Chinese with English abstract).
- [2] Zhang G, Li Y, Lin T. Caching in information centric networking: A survey. *Computer Networks*, 2013,57(16):3128–3141.
- [3] Xylomenos G, Ververidis CN, Siris VA, *et al.* A survey of information-centric networking research. *IEEE Communications Surveys & Tutorials*, 2014,16(2):1024–1049.
- [4] Wang Y, Li Z, Tyson G, *et al.* Design and evaluation of the optimal cache allocation for content-centric networking. *IEEE Trans. on Computers*, 2016,65(1):95–107.
- [5] Psaras I, Chai WK, Pavlou G. Probabilistic in-network caching for information-centric networks. In: *Proc. of the 2nd ICN Workshop on Information-centric Networking*. New York: ACM, 2012. 55–60.
- [6] Dabirmoghaddam A, Barijough MM, Garcia-Luna-Aceves JJ. Understanding optimal caching and opportunistic caching at “the edge” of information-centric networks. In: *Proc. of the 1st Int’l Conf.* New York: ACM, 2014. 47–56.
- [7] Rossi D, Rossini G. On sizing CCN content stores by exploiting topological information. In: *Proc. of the 2012 IEEE Conf. on Computer Communications Workshops (INFOCOM WKSHPS)*. Piscataway: IEEE, 2012. 280–285.
- [8] Li Y, Xie H, Wen Y, *et al.* How much to coordinate? Optimizing in-network caching in content-centric networks. *IEEE Trans. on Network & Service Management*, 2015,12(3):420–434.
- [9] Gill AS, D’Acunto L, Trichias K, *et al.* BidCache: Auction-based in-network caching in ICN. In: *Proc. of the 2016 IEEE Globecom Workshops (GC Wkshps)*. Piscataway: IEEE, 2016. 1–6.
- [10] Breslau L, Cao P, Fan L, *et al.* Web caching and Zipf-like distributions: Evidence and implications. In: *Proc. of the 18th Annual Joint Conf. of the IEEE Computer and Communications Societies (INFOCOM’99)*. Piscataway: IEEE, 1999. 126–134.
- [11] Zhang M, Luo H, Zhang H. A survey of caching mechanisms in information-centric networking. *IEEE Communications Surveys & Tutorials*, 2015,17(3):1473–1499.
- [12] Jacobson V, Smetters DK, Thornton JD, *et al.* Networking named content. In: *Proc. of the 5th Int’l Conf. on Emerging Networking Experiments and Technologies*. New York: ACM, 2009. 1–12.
- [13] Xu A, Tan X, Tian Y. Design and evaluation of a utility-based caching mechanism for information-centric networks. In: *Proc. of the 2015 IEEE Int’l Conf. on Communications (ICC)*. Piscataway: IEEE, 2015. 5535–5540.
- [14] Ren J, Qi W, Westphal C, *et al.* Magic: A distributed max-gain in-network caching strategy in information-centric networks. In: *Proc. of the 2014 IEEE Conf. on Computer Communications Workshops (INFOCOM WKSHPS)*. Piscataway: IEEE, 2014. 470–475.
- [15] Thar K, Oo TZ, Pham C, *et al.* Efficient forwarding and popularity based caching for content centric network. In: *Proc. of the 2015 Int’l Conf. on Information Networking (ICOIN)*. Piscataway: IEEE, 2015. 330–335.
- [16] Saino L, Psaras I, Pavlou G. Hash-routing schemes for information centric networking. In: *Proc. of the 3rd ACM SIGCOMM Workshop on Information-centric Networking*. New York: ACM, 2013. 27–32.
- [17] Sourlas V, Psaras I, Saino L, *et al.* Efficient Hash-routing and domain clustering techniques for information-centric networks. *Computer Networks*, 2016,103:67–83.
- [18] Wang S, Bi J, Wu J, *et al.* CPHR: In-network caching for information-centric networking with partitioning and Hash-routing. *IEEE/ACM Trans. on Networking*, 2016,24(5):2742–2755.
- [19] Li W, Li Y, Wang W, *et al.* A collaborative caching scheme with network clustering and Hash-routing in CCN. In: *Proc. of the 2016 IEEE 27th Annual Int’l Symp. on Personal, Indoor, and Mobile Radio Communications (PIMRC)*. Piscataway: IEEE, 2016. 1–7.

- [20] Zhang G, Wang X, Gao Q, *et al.* A hybrid ICN cache coordination scheme based on role division between cache nodes. In: Proc. of the 2015 IEEE Global Communications Conf. (GLOBECOM). San Diego: IEEE, 2015. 1–6.
- [21] Yu M, Li R. Dynamic popularity-based caching permission strategy for named data networking. In: Proc. of the 2018 IEEE 22nd Int'l Conf. on Computer Supported Cooperative Work in Design (CSCWD). Nanjing: IEEE, 2018. 576–581.
- [22] Yovita LV, Syambas NR., Matheus Edward IY. CAPIC: Cache based on popularity and class in named data network. In: Proc. of the 2018 Int'l Conf. on Control, Electronics, Renewable Energy and Communications (ICCEREC). Bandung: IEEE, 2018. 24–29.
- [23] Yu M, Li R, Liu Y, *et al.* A caching strategy based on content popularity and router level for NDN. In: Proc. of the 2017 7th IEEE Int'l Conf. on Electronics Information and Emergency Communication (ICEIEC). Macau: IEEE, 2017. 195–198.
- [24] Li Y, Xie H, Wen Y, *et al.* Coordinating in-network caching in content-centric networks: Model and analysis. In: Proc. of the 2013 IEEE 33rd Int'l Conf. on Distributed Computing Systems. Philadelphia: IEEE, 2013. 62–72.
- [25] Chang CY, Chang MS. A hybrid coordination approach of in-network caching for named data networking. *Int'l Journal of Future Generation Communication and Networking*, 2016,9(4):285–300.
- [26] Fayazbakhsh SK, Lin Y, Tootoonchian A, *et al.* Less pain, most of the gain: Incrementally deployable ICN. *Proc. of the ACM SIGCOMM Computer Communication Review*, 2013,43(4):147–158.
- [27] Wang Y, Li Z, Tyson G, *et al.* Optimal cache allocation for content-centric networking. In: Proc. of the 2013 21st IEEE Int'l Conf. on Network Protocols (ICNP). Piscataway: IEEE, 2013. 1–10.
- [28] Chiochetti R, Rossi D, Rossini G. CCNSIM: An highly scalable ccn simulator. In: Proc. of the 2013 IEEE Int'l Conf. on Communications (ICC). Piscataway: IEEE, 2013. 2309–2314.
- [29] <http://www.topology-zoo.org/maps/IowaStatewideFiberMap.jpg>

#### 附中文参考文献:

- [1] 王兴伟,李婕,谭振华,马连博,李福亮,黄敏.面向“互联网+”的网络技术发展现状与未来趋势. *计算机研究与发展*,2016,53(4): 729–741.



刘嘉琦(1995—),男,硕士生,主要研究领域为计算机网络.



孟绪颖(1992—),女,博士,助理研究员,CCF 学生会会员,主要研究领域为智能网络.



张亚文(1997—),女,硕士生,主要研究领域为计算机网络.



周继华(1979—),男,博士,研究员,博士生导师,主要研究领域为无线通信,移动网络,4G,5G,特种通信.



张瀚文(1981—),女,博士,副研究员,CCF 高级会员,主要研究领域为计算机网络体系结构.



张玉军(1976—),男,博士,研究员,博士生导师,CCF 高级会员,主要研究领域为计算机网络.