

文章编号:1674-2974(2016)08-0142-09

# 内容中心网络中一种基于内容扩散的主动缓存机制<sup>\*</sup>

罗熹<sup>1,3</sup>, 安莹<sup>2†</sup>, 牛碧诺<sup>4</sup>

(1. 中南大学 信息科学与工程学院, 湖南 长沙 410083; 2. 中南大学 信息安全与大数据研究院, 湖南 长沙 410083;  
3. 湖南警察学院 信息技术系, 湖南 长沙 410138; 4. 武汉大学 计算机学院, 湖北 武汉 430072)

**摘要:**借鉴分子扩散的思想,提出一种基于内容扩散的主动缓存机制(Content Diffusion Based Proactive Caching, CDBPC).该机制引入缓存内容浓度的概念来描述不同内容在不同区域内的需求程度,然后根据节点间的缓存内容浓度关系来驱动内容副本在网络中的主动推进和迁移,并结合内容的流行度等因素实现了缓存内容的概率性放置,从而达到内容缓存的快速部署和推进,提高为用户提供就近响应概率的目的.仿真结果表明,该机制能有效地降低系统的平均接入代价并提高缓存命中率.

**关键词:**内容中心网络;扩散原理;主动缓存;内容流行度

**中图分类号:**TP393

**文献标识码:**A

## Content Diffusion Based Proactive Caching in Content-centric Networks

LUO Xi<sup>1,3</sup>, AN Ying<sup>2†</sup>, NIU Bi-nuo<sup>4</sup>

(1. School of Information Science and Engineering, Central South Univ, Changsha, Hunan 410083, China;  
2. Institute of Information Security & Big Data, Central South Univ, Changsha, Hunan 410083, China;  
3. Department of Information Technology, Hunan Police Academy, Changsha, Hunan 410138, China;  
4. Computer School, Wuhan Univ, Wuhan, Hubei 430072, China)

**Abstract:** This paper borrowed the idea of molecular diffusion and proposed a proactive caching scheme based on content diffusion. In this mechanism, the conception of content concentration was introduced to analyze user demands for different contents in different network regions. In order to achieve fast content placement and increase the probability of providing responses to user requests by the nearer caching nodes, the content replicas were proactively pushed forward or migrated according to the content concentration difference between nodes. Furthermore, by synthetically considering the content popularity, a probabilistic content placement method was also implemented. The simulation results have shown that this scheme can effectively decrease the average access cost and improve the cache hit ratio.

**Key words:** Content-centric Networking(CCN); diffusion theory; proactive caching; content popularity

<sup>\*</sup> 收稿日期:2015-06-23

**基金项目:**国家自然科学基金资助项目(61402541), National Natural Science Foundation of China(61402541); 网络犯罪侦查湖南省普通高等学校重点实验室开放研究基金资助项目(2016WLFZZC009); 网络侦查技术湖南省普通重点实验室开放研究基金资助项目(2016WLZC009)

**作者简介:**罗熹(1980-),女,湖南长沙人,中南大学博士研究生,湖南警察学院讲师

<sup>†</sup> 通讯联系人, E-mail: anying@csu.edu.cn

近年来,随着 P2P 应用、发布订阅系统、普适计算以及海量流媒体等新型应用的不断发展,信息获取已经成为了当前网络服务的主体,传统的以主机为中心的通信模式逐渐演变为以信息为中心的新模式。网络通信方式的变化使得网络设计者们必须对当前的 Internet 体系架构做出重大调整,从而导致了以内容中心网络(Content-Centric Networking, CCN)<sup>[1]</sup>为代表的一系列新型网络的涌现。在 CCN 中,内容位置的重要性被逐渐弱化,网络通过对内容进行统一命名,来实现基于内容名称的定位、路由和传输。同时,CCN 要求每个具有存储能力的节点都能缓存经过的内容,从而为其他用户对同一内容的后续请求提供快速的响应。换言之,CCN 将透明化、泛在化的网络内置缓存作为其网络体系结构中固有的一部分,以此加快内容的分发速度并提高网络资源的利用率。因此,CCN 中的内置缓存机制也就成为了目前该领域的研究热点,其中一个重要的问题是如何选择内容的缓存位置,并且研究者们已经就此开展了大量的研究工作。Psaras 等人<sup>[2]</sup>提出了一种基于加权概率的缓存机制 ProbCache,结合节点与请求者的距离和路径的剩余缓存空间来计算内容返回路径上各沿途节点缓存内容对象的概率,该概率与节点和内容请求者间的距离成反比,而与节点的可用资源成正比。该机制通过将内容以更高的概率缓存于距离请求者更近的节点,来保证内容副本快速地趋向网络边缘,从而提升系统的缓存性能。文献[3-4]采用了相似的基本思想,通过为流行度高的内容设置较长的缓存时间,保证流行内容的缓存命中,并加快内容获取的响应速度。文献[5]采用了差异化缓存的方式,提出了一种结合空间存储位置和缓存驻留时间的二维差异化缓存策略。该策略将一定时间间隔内内容被请求的次数定义为该内容的活跃度,然后根据内容活跃度的变化趋势来决定下行节点是否对内容进行缓存,在空间维度上实现内容缓存位置的逐跳推进。同时,通过赋予活跃度高的内容更长的缓存时间,进一步确保流行内容尽可能地缓存在靠近用户的网络边缘位置。刘外喜等人<sup>[6]</sup>提出了把内容的放置、发现、替换统一起来考虑的 APDR 机制,实现内容的有序缓存。APDR 的主要思想是:Interest 报文除了携带对内容的请求,还收集沿途各节点对该内容的潜在需求、空闲缓存等信息,使得 Interest 的汇聚点和目的地节点,可以据

此计算出一个缓存方案,并把该方案附加在 Data 报文之上,通知返程途中的某些节点缓存该内容并设置指定的缓存时间。文献[7]根据用户的潜在需求和内容的流行规律,提出了一种选择性缓存机制 SC。作者认为由于下游节点缓存的存在,节点上收到过某个内容的请求并作出响应的端口在未来一段时间内不会再次收到对该内容的请求。因此,对于某个内容而言,节点上未请求过该内容的端口数量越少,其对该内容的潜在需求就越低,那么节点缓存该内容的必要性也越小。在此基础上,该机制通过结合链路利用率以及节点的可用缓存空间等因素计算内容的缓存概率来实现内容的选择性缓存,以提升系统的缓存效率。Kyi 等人<sup>[8]</sup>提出了一种基于一致性哈希算法的协作缓存决策机制,该机制将 AS 内的路由节点分成不同的组,通过组内路由节点间对内容缓存和请求转发的协同操作来提高缓存性能和资源利用率。在缓存决策时,各个路由节点主要考虑内容的流行度并利用一致性哈希避免内容的重复缓存。

CCN 中内容缓存的重要目的之一是逐步地使内容副本接近请求用户,从而缩短内容的获取延迟。然而,对于整个网络范围而言,内容对象向网络边缘的复制往往是一个较为缓慢的过程<sup>[9]</sup>。其主要原因是,现有的 CCN 缓存机制中,内容的缓存决策大多和内容的分发过程一样均由用户的兴趣(Interest)驱动,并采用路径内(on-path)缓存策略直接从 Interest 的反向路径上的中间节点中选择内容缓存位置。这导致了两方面的主要缺陷,其一,内容的携带者(原始内容服务器或中间缓存节点)只有在收到相应的 Interest 并在其缓存中发生命中后才会响应内容的返回过程中触发内容缓存位置的决策,缺乏对内容副本进行主动推进的能力。这种过于被动的方式显然会延缓内容缓存在网络内的部署速度,很大程度上它将受到 Interest 转发策略的影响。其二,内容缓存位置的选择往往局限在原始内容服务器与请求用户节点的直接路径上,未充分地考虑路径外的节点状态及用户需求。文献[10]指出路径外(off-path)缓存策略在能耗等缓存性能上比路径内(on-path)缓存策略具有更大的优势。更重要的是,如果能根据节点未来的潜在需求,预先将内容复制到当前路径以外的其他网络区域,将能更进一步地加快内容查询和获取的速度,大大提升缓存系统的综合性能。

针对上述问题,本文提出一种基于扩散理论的

主动缓存机制. 该机制不仅能根据收到的用户请求沿着反向路径选择缓存内容的放置位置, 同时还通过借鉴分子动理论中的扩散原理, 根据节点分布密度以及与内容提供者与请求者间的距离来定义网络中不同区域的缓存内容浓度, 然后将内容副本由高浓度区域主动扩散到低浓度区域, 实现内容缓存节点的快速部署以及缓存位置的动态调整, 从而有效提高系统的缓存性能. 根据现有的相关工作, 部分学者也试图通过引入物理场的概念来优化 CCN 中的缓存决策问题. 如文献[11]中提出了一种基于势能的目标识别路由方法(Cache Aware Target iden Tification, CATT), 该方法引入势场(potential field)的概念, 将节点存储的内容副本沿势能由高到低的方向进行局部范围通告, 从而提高缓存资源的可用性. 文献[12]指出 CCN 中能够响应内容请求的节点可以分为持久稳定的原始内容服务器和动态可变的临时缓存节点两类, 考虑到它们的差异化特征并结合数据场的思想, 分别为其构建全局导向和局部吸引的势能辐射场, 在路由查找过程中引导内容请求的转发, 实现缓存内容整体利用率的提升. 文献[13]则通过建立一个基于内容的拓扑势场, 用于扩散网络缓存副本的内容通告, 使网络后续产生的用户请求可以充分利用不在路径上的内容副本来选择最佳的内容响应位置. 然而这些方法均从优化内容副本通告或用户请求转发的角度来改进内容的缓存性能和网络通信服务质量. 与之不同的是, 我们提出的方法是通过构建内容在网络中的区域缓存浓度, 利用内容副本在不同浓度区域间的扩散实现内容缓存的主动放置和动态迁移, 从而增加为潜在用户请求提供就近响应的可能. 这种内容副本的主动推进方式将大大地缩短内容查找和响应的延迟并提高内容的缓存命中概率.

## 1 分子扩散的引入

扩散现象是指物质分子从高浓度区域向低浓度区域转移直到均匀分布的现象. 根据分子动理论, 分子扩散的本质是由于分子热运动而产生的质量迁移现象. 从微观上说, 它实际上是大量物质分子做无规则热运动时, 分子之间发生相互碰撞的结果. 由于不同空间区域的分子密度分布不均匀, 分子发生碰撞的情况也不同. 这种碰撞迫使密度大的区域的分子向密度小的区域转移, 最后达到均匀的密度分布.

扩散定律描述了分子扩散导致的传质过程, 这

一定律由德国物理学家菲克(A. Fick)于1855年在实验的基础上提出的, 故又称为菲克(Fick)定律. Fick 定律指出, 在稳态扩散过程中, 扩散通量  $J$  与浓度梯度成正比, 即

$$J = -D \frac{\partial \varphi}{\partial x}. \quad (1)$$

式中:  $D$  为扩散系数, 是描述扩散速度的重要物理量, 它表示单位浓度梯度条件下, 单位时间单位截面上通过的物质流量;  $\varphi$  为浓度;  $x$  代表位置; 负号表示物质沿着浓度降低的方向扩散. 式(1)的意思是, 如果某个事物的空间分布是不均匀的, 就会造成流动, 而引起的物质流正比于该事物的梯度. 其中, 梯度是物质流动的驱动力.

受到上述分子扩散思想的启发, 我们将内容对象看作溶质, 将用户对该内容对象的需求视为溶剂, 那么整个网络系统则为内容对象与用户需求混合而成的溶液. 由于内容副本以及用户需求分布的不均匀性和动态性, 因此该溶液不同区域的浓度也存在差异, 内容原始服务器或缓存节点附近浓度较高, 而内容请求节点附近浓度较低. 对于内容的缓存决策而言, 直觉上内容副本应尽量趋近用户需求较大的区域以提供快速高效的缓存和通信服务. 于是, 缓存内容的放置决策可以看作溶质分子(内容副本)在溶液中由高浓度区域向低浓度区域转移的扩散过程.

## 2 CDBPC 缓存机制

### 2.1 内容扩散模型

假设网络中存在内容集合  $C = \{c_i \mid i = 1, 2, \dots, k\}$ ,  $C$  中的元素  $c_i$  按其流行度级别编号, 其被访问的概率为  $p_i$ , 对任意  $1 \leq i < j \leq k$ , 有  $p_i \leq p_j$ . 将  $p_i$  称为内容  $c_i$  在内容集合  $C$  中的流行度, 而称  $P = \{p_i \mid i = 1, 2, \dots, k\}$  为  $C$  的流行度分布. 根据已有的研究工作, 内容的流行度服从 Zipf 或 Zipf-like 分布<sup>[14]</sup>, 因此,  $C$  中任意元素  $c_i$  的流行度可以表示为:

$$p_i = \Omega / i^\alpha. \quad (2)$$

式中:  $\Omega = \left( \sum_{i=1}^k \frac{1}{i^\alpha} \right)^{-1}$ .

将具有不同流行度的内容比作不同种类的溶质, 每类内容的一个副本即为该类溶质的一个分子. 那么某类内容  $c_i$  的缓存放置过程可以视为溶质  $c_i$  的分子在溶剂中的扩散过程. 类比扩散原理中的相关概念, 我们重新定义了扩散系数、缓存内容浓度和缓存概率 3 个概念, 表述如下.

**定义 1** 扩散系数  $D$ . 物质的分子扩散系数表示它的扩散能力, 是物质的物理性质之一. 根据菲克定律, 扩散系数是沿扩散方向, 在单位时间每单位浓度梯度的条件下, 垂直通过单位面积所扩散某物质的流量. 在内容缓存过程中, 内容的流行度反映了用户对其需求程度的高低. 高流行度的内容需要更多的缓存副本来满足用户更高的需求量. 即内容的流行度越高, 相应地它应具有更强的扩散能力, 从而保证其缓存副本能够更快更密集地分布到网络中. 因此, 本文用流行度来表示内容  $c_i$  的扩散系数, 即

$$D_i = p_i = \Omega/i^\alpha. \quad (3)$$

**定义 2** 缓存内容浓度 (Content Concentration, CC). 给定某个节点  $N$  以及内容  $c_i$ , 本文中节点  $N$  处内容  $c_i$  的缓存内容浓度  $CC_N^{c_i}$  定义为节点  $N$  所在位置附近  $c_i$  副本满足相应用户需求的程度. 若将  $c_i$  的原始内容服务器或缓存节点处的缓存内容浓度定义为最高值 1, 而将  $c_i$  请求节点处的浓度定义为最小值 0. 对于任意中间节点  $N$  所在位置,  $c_i$  的缓存内容浓度应该与  $N$  到  $c_i$  最近的缓存副本处的距离  $d_{NC}$  成反比, 而与  $N$  到  $c_i$  最近的请求节点处的距离  $d_{NR}$  成正比. 同时, 考虑到对于某个内容而言, 一定区域内用户节点的数量越多, 将来该区域内的用户中产生对该内容的请求的潜在可能性也越高. 因此, 用节点  $N$  的度中心性  $DC_N$  表示其所在区域的节点密度, 它将与缓存内容浓度成反比. 那么,  $CC_N^{c_i}$  可以表示为:

$$CC_N^{c_i} = \begin{cases} e^{-\frac{d_{NC}}{d_{NR}} DC_N}, & d_{NR} > 0; \\ 0, & d_{NR} = 0. \end{cases} \quad (4)$$

## 2.2 CDBPC 工作原理

CDBPC 机制的基本思想是克服现有缓存机制中缓存决策过程完全依赖用户 Interest 驱动的被动性, 通过借鉴分子扩散的原理以一种更主动的方式实现内容副本向用户端的快速推进. 同时在内容副本的推进过程中综合缓存内容浓度和内容流行度实现概率性的缓存放置, 从而加快对后续用户请求的响应速度, 提高系统的整体缓存性能. CDBPC 主要包括两个部分: 内容副本的推进以及内容的缓存决策. 在内容副本的推进过程中, 我们设计了被动响应和主动扩散 2 种基本的推进模式. 其中, 被动响应模式发生在节点返回被请求的数据的响应过程中. 即, 仅当收到相应的用户请求并在缓存中发生命中时, 作为数据应答, 节点才会将内容副本沿着请求的反向传输路径返回请求用户. 主动扩散模式的触发则

与内容是否在该节点发生缓存命中无关, 而是每隔一定的时间 (本文称之为扩散周期  $T_d$ ), 节点便通过计算并比较自身与邻居节点的缓存内容浓度, 主动将内容副本沿着缓存内容浓度下降最快的路径方向逐跳地进行扩散. 在内容副本的主动扩散或被动响应路径上, 中间节点再以一定的概率来选择内容的缓存位置. CDBPC 机制的工作流程描述如下.

1) 每个节点包含一个计时器  $T$ , 在网络初始化时,  $T$  被清零并开始计时. 若任意节点  $N$  的缓存内容集合  $C$  非空 (即节点至少缓存了一个内容), 则  $N$  依据内容的流行度以一定的概率从中挑选出一个内容  $c_i$  作为待处理对象.

2)  $N$  按照公式 (4) 计算当前节点位置  $c_i$  的缓存内容浓度  $CC_N^{c_i}$ , 并与各邻居进行比较. 若  $CC_N^{c_i}$  大于邻居节点处的缓存内容浓度, 则  $N$  从中选择缓存内容浓度差最大的邻居节点按照浓度由高到低的方向对  $c_i$  的副本进行主动扩散, 同时将  $N$  的计时器清零. 否则, 不进行主动扩散.

3) 每经过一个扩散周期  $T_d$ , 节点  $N$  将重复步骤 1) - 2) 决定是否执行内容的主动扩散.

4) 当  $N$  收到某个内容副本  $c_j$ , 则  $N$  根据概率性缓存决策策略以一定的概率缓存该内容副本, 同时检查其待定转发表 PIT.

5) 若在 PIT 中发现与  $c_j$  匹配的用户请求, 则  $N$  采用被动响应模式, 根据 PIT 中记录的请求来源端口将  $c_j$  返回请求用户节点. 否则以  $c_j$  作为待处理对象, 执行步骤 2) 决定是否进行内容的主动扩散.

6) 若  $c_j$  未被节点  $N$  缓存, 同时也未触发相应的主动扩散或被动响应, 则  $N$  直接丢弃  $c_j$ .

这里, 我们的主要目的是利用内容的主动扩散驱动内容副本向用户端的预先推进和部署, 来减小用户进行内容查找以及获取的延时. 下面将具体介绍 CDBPC 机制的主要实现细节.

### 2.2.1 缓存内容浓度的计算

在 CDBPC 机制中, 内容副本的主动扩散以及缓存概率均与缓存内容浓度有着密切关系, 而根据公式 (4), 节点  $N$  处内容  $c_i$  的缓存内容浓度  $CC_N^{c_i}$  由  $N$  到  $c_i$  最近的缓存副本的距离  $d_{NC}$  和  $N$  到  $c_i$  最近的请求节点的距离  $d_{NR}$  决定. 然而, 由于在实际网络中内容请求节点的位置通常无法预知, 再加上内容缓存位置的不确定性, 导致  $d_{NC}$  和  $d_{NR}$  是动态变化的, 因此,  $CC_N^{c_i}$  的值也需要根据  $d_{NC}$  和  $d_{NR}$  的变化来实现动态更新. 显然, 存储了  $c_i$  的节点 (其原始内容服务器或中间缓存节点) 处  $d_{NC}$  值为 0, 而  $c_i$  请求节点

处  $d_{NR}$  的值则为 0. 其他情况下, 将节点的  $d_{NC}$  和  $d_{NR}$  初值均设为某个较大的值(如网络节点总数). 为了获得与内容请求节点和内容存储节点的距离信息, 我们在用户的 Interest 包和内容的 Data 包中各增加一个跳数字段, 分别记录其从源节点到当前位置经历的转发跳数. 当节点收到某个 Interest 包时, 将会比较该包头中记录的跳数值与其当前的  $d_{NR}$  值, 然后将  $d_{NR}$  更新为二者中的较小值. 同理, 节点根据经过的 Data 包中的跳数值来实现对  $d_{NC}$  值的更新. 这样, 每个节点即可按照公式(4)计算自身位置处指定内容的缓存内容浓度, 并在自己的  $d_{NC}$ ,  $d_{NR}$  或度中心性值发生变化时及时对其进行更新.

### 2.2.2 内容的主动扩散

在 CDBPC 中, 被动响应模式相对比较简单, 它由发生缓存命中的 Interest 包触发, 内容副本按照预先确定的 Interest 包的反向路径推进. 而在主动扩散模式中, 内容副本的扩散源自相邻节点间的缓存内容浓度差, 按浓度由高到低的方向进行移动. 与被动响应模式中推进的内容对象以及推进的路径均由相应的 Interest 确定不同, 主动扩散模式需解决以下 2 个问题: ①节点缓存中可能存在多个不同的内容对象, 对所有的缓存内容均进行扩散容易造成过大的通信和存储开销. 因此必须对进行扩散的内容对象进行合理的选择; ②沿着节点的所有输出路径进行全方位的扩散也必然产生大量的冗余内容副本, 导致网络资源的过度消耗. 因此, 节点还必须合理地选择内容副本的推进方向以实现内容的选择性扩散. 下面针对上述问题给出具体的解决策略.

1) 扩散内容的选择. 为了使不同的内容对象均能获得一定的扩散机会, 保证不同内容间的公平性以及网络中缓存内容的多样性, 我们采用概率性选择的方式从节点的缓存中确定待扩散的内容. 考虑到用户对流行度高的内容往往具有更高的请求概率, 因此我们根据内容的流行度来调整不同内容的选中概率, 使其与各自的流行度成正比. 具体来说, 假设节点  $N$  的缓存中包含  $l$  个内容对象, 若将其中的第  $i$  个内容对象表示为  $c_i^N$ , 其流行度表示为  $p_{c_i^N}$ . 那么,  $c_i^N$  被选中为扩散内容对象的概率  $DP_i$  可以按公式(5)计算得到.

$$DP_i = p_{c_i^N} / \sum_{i=1}^l p_{c_i^N}. \quad (5)$$

2) 扩散方向的选择. 为了控制内容扩散的传输开销, CDBPC 中节点通过比较自己与各邻居间的缓存内容浓度, 每次仅选择向浓度下降最快的邻居节

点方向进行扩散. 以节点  $S$  为例, 假设其邻居节点集合为  $\text{Neighbor} = \{N_1, N_2, \dots, N_n\}$ , 选中的扩散内容对象为  $c_i$ , 那么节点  $S$  与任意邻居节点  $N_j$  间的缓存内容浓度差可以表示为:

$$\Delta CC_{Sj}^{c_i} = CC_{Sj}^{c_i} - CC_S^{c_i}. \quad (6)$$

如果  $S$  的邻居中存在节点  $N_k$ , 满足  $\Delta CC_{Sj}^{c_i} = \max\{\Delta CC_{Sj}^{c_i} \mid j = 1, 2, \dots, n\}$ , 且  $\Delta CC_{Sj}^{c_i} > \theta_{th}$ , 则节点  $S$  选择将  $c_i$  向节点  $N_k$  进行扩散. 否则,  $S$  将不会触发对  $c_i$  的主动扩散. 这里,  $\theta_{th}$  称为扩散浓度差阈值, 它是一个大于 0 的正数, 表示触发主动扩散所需的节点间最小缓存内容浓度差. 扩散浓度差阈值的作用是避免在节点间缓存内容浓度较为接近时触发内容的主动扩散而引起节点间浓度关系来回摆动, 导致算法无法达到收敛的问题.

### 2.2.3 概率性内容缓存

CDBPC 机制中, 内容副本推进路径上的节点将以一定的概率对内容副本进行缓存. 在确定不同节点对内容副本的缓存概率时, 我们主要考虑两方面的因素. 首先, 从内容本身的角度来看, 内容的流行度越高, 其被用户请求的概率也就越大. 因此, 节点应该以更高的概率来缓存流行度高的内容. 另一方面, 从节点的角度来看, 将内容缓存在距离请求用户节点越近的位置, 将能获得更快的响应速度以及更高的缓存命中率. 因此, 内容副本推进路径上各节点的缓存概率应该沿着内容副本的推进方向逐跳递增, 而缓存概率增加的比例则应与内容推进方向上缓存内容浓度下降的速度成正比.

具体来说, 假设节点  $S$  为内容  $c_i$  的原始内容服务器或缓存节点,  $c_i$  的流行度为  $p_{c_i}$ .  $S$  沿着路径  $\text{path} = \{N_1, N_2, N_3, \dots, N_k, \dots\}$  对内容  $c_i$  进行推进, 其中  $N_k$  表示  $c_i$  推进路径上的第  $k$  跳中间节点. 若节点  $N_k$  处内容  $c_i$  的缓存内容浓度为  $CC_k^{c_i}$ ,  $S$  与第一跳节点  $N_1$  的缓存内容浓度差为  $\Delta CC_{S1}^{c_i}$ , 那么  $N_k$  缓存  $c_i$  的概率可表示为:

$$CP_k^{c_i} = \min\left\{p_{c_i} \frac{1 - CC_k^{c_i}}{\Delta CC_{S1}^{c_i}}, 1\right\}. \quad (7)$$

需要说明的是, 若  $N_k$  成为了内容  $c_i$  的新的缓存节点, 那么, 将以  $N_k$  作为新的起始节点, 而原路径中  $N_k$  之后的子路径即为  $c_i$  新的推进路径, 然后利用公式(7)重新计算新路径上各节点对  $c_i$  的缓存概率并执行概率性缓存决策.

## 3 性能仿真及评价

本文采用 ndnSIM<sup>[15]</sup> 对算法进行了仿真和性

能分析. 仿真实验中我们利用 GT-ITM 下的 Locality 模型构建了一个由 2 个内容服务器和 50 个路由节点组成的平面随机网络拓扑. 其中, 内容服务器位于网络的边缘, 并从边缘节点中随机选择 2 个节点与之相连. 2 个内容服务器各自不重复地随机存储占网络内容对象总数一半的内容. 同时, 假设用户请求的到达过程服从  $\lambda=50$  的泊松分布, 用户的访问模式服从参数为  $\alpha$  的 Zipf 分布. 各路由节点缓存容量相同, 且初始状态下节点缓存为空. 请求分组采用洪泛方式进行转发, 默认缓存替换策略为 LRU, 其他的主要实验参数如表 1 所示.

表 1 实验参数设置  
Tab. 1 Parameter settings

主要参数	默认值	变动范围
内容数量/个	5 000	1 000~10 000
内容大小/MB	1	—
节点缓存/MB	50	10~100
用户访问模式	Zipf: $\alpha=0.7$	0.1~1.0
仿真时间/s	200	—

为了对算法性能进行评价, 我们选择了缓存命中率和平均接入代价作为主要的性能指标.

1) 缓存命中率. 缓存命中率是评价缓存性能的重要指标之一, 它被定义为用户请求由缓存而非原始内容服务器响应的概率. 缓存内容的命中率越高, 对应的原始内容服务器响应率和负载就越小, 系统的效率就越高.

2) 平均接入代价. 平均接入代价被定义为用户请求找到匹配的内容对象所需的平均跳数, 它反映了用户请求得到响应, 用户完成内容获取的速度. 平均接入代价越小, 内容获取速度越快, 缓存系统效率越高.

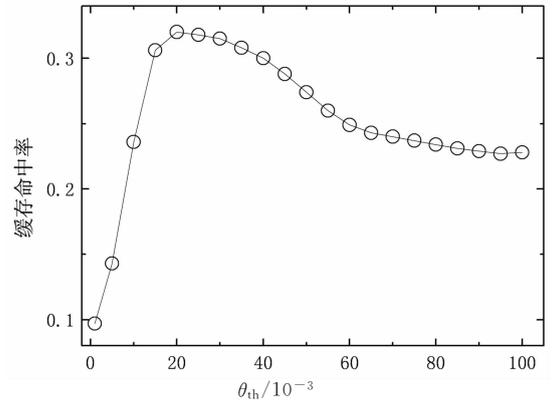
### 3.1 CDBPC 参数选择

CDBPC 包含两个主要的算法参数: 扩散浓度差阈值  $\theta_{th}$  和扩散周期  $T_d$ , 本小节将通过仿真实验来选择合理的参数值.

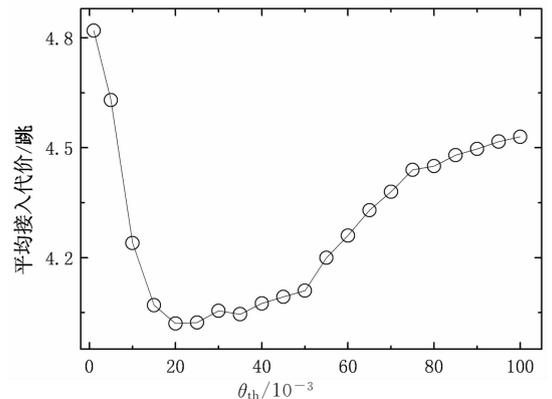
#### 3.1.1 扩散浓度差阈值 $\theta_{th}$

缓存内容浓度与扩散浓度差阈值  $\theta_{th}$  的大小关系是 CDBPC 机制中内容副本主动扩散的触发条件. 较小的  $\theta_{th}$  值意味着节点对内容副本进行主动扩散的概率越大, 内容副本在网络中扩散的范围也相对更大, 从而加速副本在网络中的推进和分布. 但是扩散主动性的增加也会带来更大的开销.  $\theta_{th}$  值过小可能造成内容副本过度扩散并缓存, 节点缓存的过快消耗必然引发大量的缓存替换操作, 反而导致缓

存命中率下降并影响了 CDBPC 机制加快内容响应速度的效果. 反之,  $\theta_{th}$  值越大, 推进内容副本的主动性越低, 扩散范围也越小, CDBPC 机制对缓存性能的积极作用逐渐减小并退化为被动响应模式. 图 1 (a) 和 (b) 分别反映了  $\theta_{th}$  对缓存命中率和平均接入代价的影响. 由图 1 可见,  $\theta_{th}$  值在 0.02 附近时 CDBPC 获得了相对较高的缓存命中率和较小的平均接入代价. 而当  $\theta_{th}$  值向两侧变化时, 则其性能均出现了不同程度的下滑. 当  $\theta_{th}$  大于 0.02 之后, CDBPC 的主动扩散作用随着  $\theta_{th}$  值的不断增长不断减少, 但最终其性能仍能与采用被动响应模式时接近. 然而随着  $\theta_{th}$  值从 0.02 继续不断减小, 缓存命中率出现急剧下降, 同时平均接入代价也迅速增加. 这正是因为过于激进的副本扩散加剧了节点的缓存消耗, 进而引发频繁的缓存替换, 大大缩短了内容副本在节点上的缓存服务时间, 因此最终降低了 CDBPC 的缓存性能. 根据仿真实验结果及上面的分析, 本文中 will 将 CDBPC 的扩散浓度差阈值设为 0.02.



(a) 缓存命中率与  $\theta_{th}$  关系曲线



(b) 平均接入代价与  $\theta_{th}$  关系曲线

图 1  $\theta_{th}$  对缓存性能的影响

Fig. 1 Impact of  $\theta_{th}$  on caching performance

#### 3.1.2 扩散周期 $T_d$

CDBPC 机制中, 内容副本的携带节点根据与邻居的缓存内容浓度差周期性地执行主动扩散过程,

扩散周期  $T_d$  是影响 CDBPC 机制性能的重要因素之一。为了合理地选择扩散周期  $T_d$  的长度,我们设置扩散浓度差阈值  $\theta_{th}=0.02$ ,其他实验参数均采用默认值,然后通过仿真实验观察  $T_d$  取不同值时 CDBPC 机制缓存性能的变化情况,如图 2 所示。当  $T_d$  值设置较小时,将使得节点频繁地进行缓存内容浓度的计算和比较并可能造成内容副本的过度扩散,这极大地增加了 CDBPC 机制的网络开销并导致频繁的缓存替换,从而降低了其整体的缓存性能。由图 2(a)可见, $T_d=10$  ms 时的缓存命中率仅为 29.1%,在  $T_d$  值增加到 100 ms 的过程中,CDBPC 的缓存命中率逐步提高到了 32%。但是,随着  $T_d$  值的继续增大,缓存命中率随后则发生明显的下降,在  $T_d=300$  ms 时缓存命中率跌至 27.9%。这主要是因为  $T_d$  值继续增大时,节点主动扩散的频率进一步降低,减慢了内容副本在网络中的推进和缓存部署速度,主动扩散的积极作用也因此被极大地削弱。出于同样的原因,图 2(b)中,CDBPC 的平均接入代价则出现了先减后增的变化趋势,同样也是在  $T_d=100$  ms 附近发生转折并达到最佳值。因此,综合图 2 的仿真结果,本文中扩散周期  $T_d$  的长度被设为 100 ms。

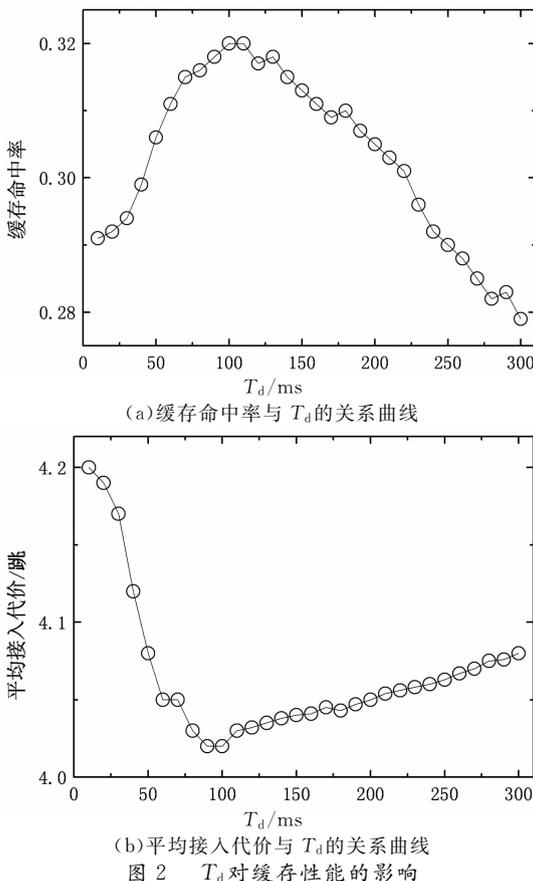


图 2  $T_d$  对缓存性能的影响  
Fig. 2 Impact of  $T_d$  on caching performance

### 3.2 性能评估

仿真过程中,我们将 CDBPC 与在用户请求的被动响应路径上选择最大介数节点进行缓存的 Betw<sup>[16]</sup> 和利用势场主动进行局部内容通告的 CATT 进行比较。其中,CATT 的内容通告范围设置为 2 跳。在缓存大小、内容数量和 Zipf 参数  $\alpha$  等网络参数变化的情况下,分别对上述 3 种缓存机制的缓存命中率以及平均接入代价进行了定量分析和比较。

#### 3.2.1 缓存大小的影响

图 3 为节点缓存对缓存性能影响的仿真结果图。由图 3(a)可以看到,随着节点缓存空间的增大,内容分组在缓存中的驻留时间延长,因此 3 种机制的缓存命中率均逐渐提高。其中,由于 CDBPC 向用户端对内容分组实施主动扩散,同时在此过程中根据缓存内容浓度和内容流行度采用了逐跳递增的概率缓存,将内容的缓存副本快速有效地分布到请求用户节点附近,从而获得了 3 种机制中最高缓存命中率。而 Betw 中内容副本的推进完全依赖节点对用户请求的应答,且单纯地依据节点的介数中心性来实现缓存节点的选择,使得介数较高的节点由于缓存压力过大而导致缓存替换频繁,因此获得的缓存命中率最低。CATT 利用势场实现了缓存内容的主动通告,有效地将用户请求导向匹配的内容缓存副本,一定程度上提高了系统的缓存命中率,因此在图中 CATT 的缓存命中率略高于 Betw。

同样,随着节点缓存的增加,中间节点对内容的缓存能力增强,每个内容分组能获得更长时间的缓存服务。这意味着用户有更大的可能从距离较近的中间缓存节点实现快速的内容获取。因此,图 3(b)中各机制的平均接入代价均随着节点缓存的增加而逐渐降低。其中,得益于 CDBPC 对内容副本的主动扩散,加快了缓存副本向用户边缘的推进,因此,CDBPC 获得了 3 种机制中最小平均接入代价。以节点缓存为 50 MB 时的情况为例,Betw 和 CATT 的平均接入代价达到了 4.55 跳和 4.22 跳,而 CDBPC 仅为 4.02 跳,与前两者相比,分别减少了近 11.6% 和 4.7%。

#### 3.2.2 内容数量的影响

内容数量对 3 种缓存机制影响的实验结果如图 4 所示。由于在节点缓存大小固定的情况下,内容数量的增加意味着可用缓存资源的相对紧张。因此,内容数量对缓存性能的影响实际上也是节点缓存大小与缓存性能之间关系的另一种体现。

从图 4(a)中结果可以看到,随着内容数量的增

多,节点缓存资源越发紧缺,3种机制的缓存命中率均出现明显下降的趋势.但是,CDBPC始终保持了3种机制中最高的缓存命中率.在内容数量增至10 000个时,CDBPC仍获得了21.6%的缓存命中率,优于CATT的19.5%和Betw的15.7%.而由图4(b)可知,3种缓存机制的平均接入代价则随着内容数量的增加逐渐增大.其中,CDBPC的平均接入代价明显低于其他2种缓存机制.同样以10 000个内容时的情况为例,CDBPC的平均接入代价仅为4.57跳,相比Betw和CATT分别减少了约10.9%和5%.这与3.2.1节中关于缓存大小对2种缓存性能指标的影响的分析结果是一致的.

同时,还得益于在内容推进以及缓存决策中对内容流行度的考虑,CDBPC在采用不同的 $\alpha$ 值时均获得了明显高于其他2种缓存机制的缓存命中性能.在平均接入代价方面,从图5(b)中的结果可以发现,用户偏好性的增强导致了平均接入代价的下降.而CDBPC对内容副本更为主动的推进方式,使得用户可以更快地实现内容获取,从而获得了3种机制中相对最低的平均接入代价.

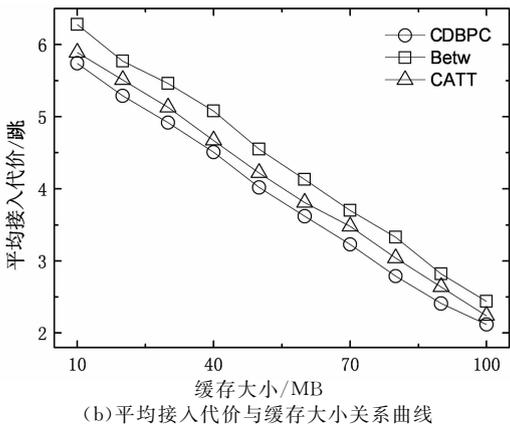
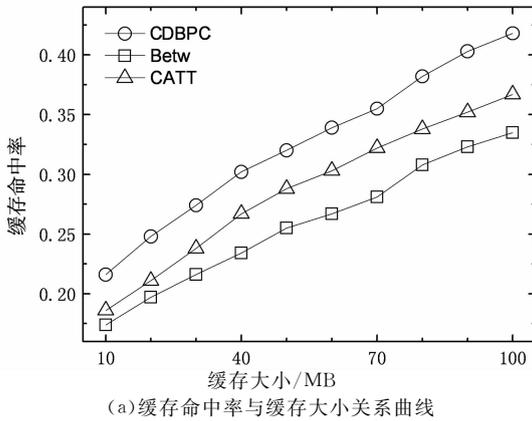


图 3 节点缓存对缓存性能的影响  
Fig. 3 Impact of cache size on caching performance

### 3.2.3 Zipf 参数 $\alpha$ 的影响

用户对内容的访问具有一定的偏好性,用户偏好对不同机制的缓存命中率的影响如图5所示. Zipf 参数  $\alpha$  越大,意味着用户的偏好越发地向流行度高的内容集中.由于3种缓存机制均采用了优先保证高流行度内容缓存时间的相关策略(如,基于LRU的缓存替换策略),因此,由图5(a)可见,它们的缓存命中率随着 $\alpha$ 值的增大均呈现上升的趋势.

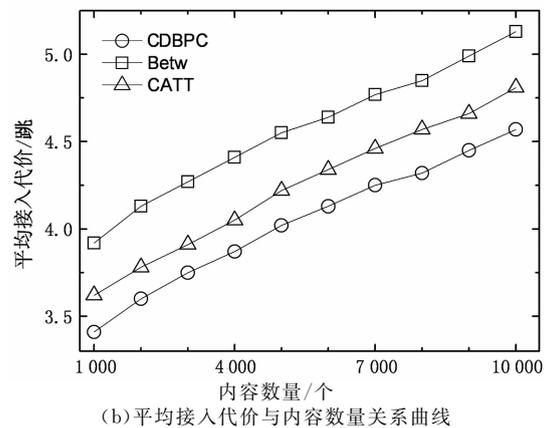
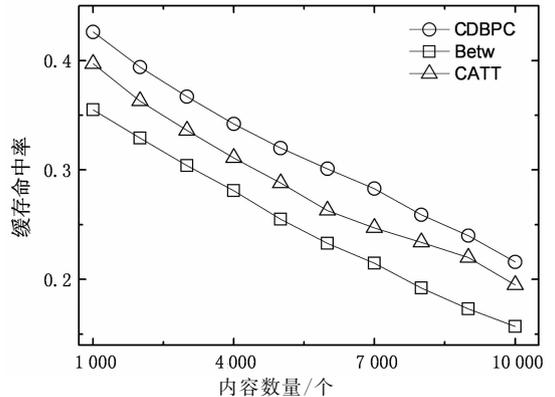
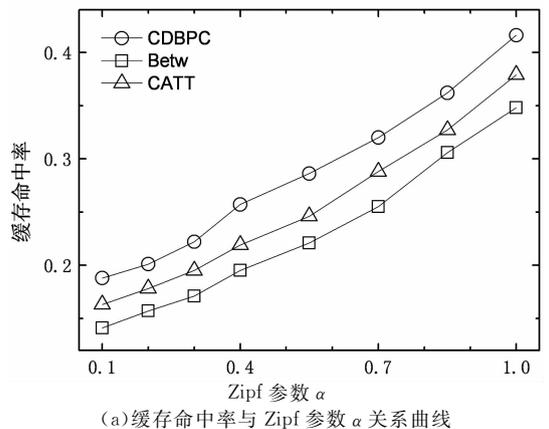
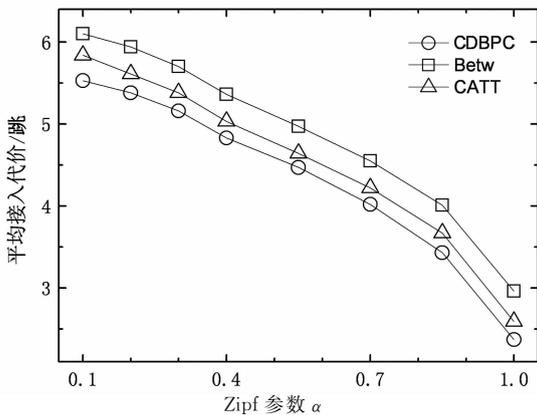


图 4 内容数量对缓存性能的影响  
Fig. 4 Impact of content number on caching performance





(b) 平均接入代价与 Zipf 参数  $\alpha$  关系曲线

图 5 Zipf 参数  $\alpha$  对缓存性能的影响

Fig. 5 Impact of Zipf parameter  $\alpha$  on caching performance

## 4 结 论

为了加快内容缓存副本在内容中心网络中的推进,进一步提高其为潜在用户请求提供就近响应的可能性,本文提出缓存内容浓度的概念,并借鉴分子的扩散原理设计了一种基于内容扩散的主动缓存机制 CDBPC. 该机制利用节点间的缓存内容浓度关系来驱动内容副本在网络中的主动推进和迁移,并结合内容的流行度等因素实现了内容的概率性缓存决策. 仿真实验结果表明该机制能有效地提高系统的整体缓存性能. 在今后的工作中,我们将对 CDBPC 在实际网络环境下的性能进行验证,同时进一步研究如何将其扩展到移动网络以及其他复杂的网络环境.

## 参考文献

- [1] JACOBSON V, SMETTERS D K, THORNTON J D, *et al.* Networking named content[J]. *Communications of the ACM*, 2012, 55(1):117-124.
- [2] PSARAS I, CHAI W K, PAVLOU G. Probabilistic in-network caching for information-centric networks[C]// *Proceedings of the Second Edition of the ICN Workshop on Information-centric Networking*. New York: ACM, 2012: 55-60.
- [3] MING Z, XU M, WANG D. Age-based cooperative caching in information-centric network[C]// *Proceedings of the 23rd International Conference on Computer Communication and Networks (ICCCN)*. Washington DC: IEEE Computer Society, 2014:1-8.
- [4] QIAN H, MUQING W, DONGYANG W, *et al.* Lifetime-based greedy caching approach for content-centric networking[C]// *Proceedings of the 21st International Conference on Telecommunications (ICT)*. Washington DC: IEEE Computer Society, 2014:426-430.

- [5] 葛国栋, 郭云飞, 刘彩霞, 等. 命名数据网络中基于内容请求相关性的协作缓存算法[J]. *电子与信息学报*, 2014, 36(12): 2795-2801.  
GE Guo-dong, GUO Yun-fei, LIU Cai-xia, *et al.* Collaborative caching algorithm based on request correlation in named data networking[J]. *Journal of Electronics & Information Technology*, 2014, 36(12):2795-2801. (In Chinese)
- [6] 刘外喜, 余顺争, 蔡君, 等. ICN 中的一种协作缓存机制[J]. *软件学报*, 2013, 24(8):1947-1962.  
LIU Wai-xi, YU Shun-zheng, CAI Jun, *et al.* Scheme for cooperative caching in ICN[J]. *Journal of Software*, 2013, 24(8): 1947-1962. (In Chinese)
- [7] 刘外喜, 余顺争, 胡晓, 等. CCN 中选择性缓存机制的研究[J]. *计算机学报*, 2014, 37(2):275-288.  
LIU Wai-xi, YU Shun-zheng, HU Xiao, *et al.* Selective caching in content-centric networking[J]. *Chinese Journal of Computers*, 2014, 37(2):275-288. (In Chinese)
- [8] KYI T, SAEED U, CHOONG S H. Consistent hashing based cooperative caching and forwarding in content centric network [C]// *Proceedings of the 16th Asia-Pacific Network Operations and Management Symposium (APNOMS)*. Washington DC: IEEE Computer Society, 2014:1-4.
- [9] 张国强, 李杨, 林涛, 等. 信息中心网络中的内置缓存技术研究[J]. *软件学报*, 2014, 25(1):154-175.  
ZHANG Guo-qiang, LI Yang, LIN Tao, *et al.* Survey of in-network caching techniques in information-centric networks[J]. *Journal of Software*, 2014, 25(1):154-175. (In Chinese)
- [10] DRAXLER M, KARL H. Efficiency of on-path and off-path caching strategies in information centric networks[C]// *Proceedings of 2012 IEEE International Conference on Green Computing and Communications (GreenCom)*. Washington DC: IEEE Computer Society, 2012:581-587.
- [11] EUM S, NAKAUCHI K, MURATA M, *et al.* CATT: potential based routing with content caching for ICN[J]. *IEEE Communications Magazine*, 2012, 50(12):60-67.
- [12] 葛国栋, 郭云飞, 刘彩霞, 等. 内容中心网络中基于差异化缓存通告的混合路由机制[J]. *电子与信息学报*, 2015, 37(3):700-707.  
GE Guo-dong, GUO Yun-fei, LIU Cai-xia, *et al.* A hybrid routing scheme based on differentiated cache advertisement in content centric networking[J]. *Journal of Electronics & Information Technology*, 2015, 37(3):700-707. (In Chinese)
- [13] 陈璐, 汤红波, 郑林浩. 内容中心网络基于拓扑势的请求者移动性支持机制[J]. *计算机工程与设计*, 2014, 35(8): 2685-2690.  
CHEN Lu, TANG Hong-bo, ZHENG Lin-hao. Consumer mobility scheme in content centric networks based on topological potential[J]. *Computer Engineering and Design*, 2014, 35(8): 2685-2690. (In Chinese)
- [14] KIM Y, YEOM I. Performance analysis of in-network caching for content-centric networking[J]. *Computer Networks*, 2013, 57(13):2465-2482.
- [15] MASTORAKIS S, AFANASYEV A, MOISEENKO I, *et al.* ndnSIM 2.0: a new version of the NDN simulator for NS-3 [R]. Los Angeles: University of California, 2015.
- [16] CHAI W K, HE D, PSARAS I, *et al.* Cache "Less for more" in information-centric networks[J]. *Computer Communications*, 2013, 36(7):758-770.