



专题:信息中心网络(ICN)

NDN 中基于蚁群替换算法的邻居协作缓存管理策略^{*}

董利利¹, 王 勇¹, 董永强^{1,2}, 杨 鹏^{1,2}

(1. 东南大学计算机科学与工程学院 南京 211189;

2. 东南大学计算机网络和信息集成教育部重点实验室 南京 211189)

摘 要:为提高 NDN(命名数据网络)中的缓存利用率,提出了一种基于蚁群替换算法的邻居协作缓存管理(ACNCM)策略。首先将单节点的缓存替换问题,建模为 0/1 背包问题,并根据缓存数据的大小、使用频率以及邻居副本深度等信息定义本地存储内容的缓存价值,提出基于蚁群算法的缓存替换算法。然后利用邻域协作的思想,通过路由节点之间定期交换自身节点的缓存信息,对单个节点替换出去的缓存内容,选择邻居节点完成协作式缓存管理。实验结果表明,ACNCM 策略在缓存命中率、网络开销和平均响应时延方面均优于现有方法。

关键词:命名数据网络;缓存管理;内容缓存价值;蚁群替换算法

doi: 10.3969/j.issn.1000-0801.2014.09.007

Collaborative Cache Management Strategy Based on Ant-Colony Replacement Algorithm in Named Data Networking

Dong Lili¹, Wang Yong¹, Dong Yongqiang^{1,2}, Yang Peng^{1,2}

(1. School of Computer Science and Engineering, Southeast University, Nanjing 211189, China;

2. Key Laboratory of Computer Network and Information Integration, Ministry of Education, Southeast University, Nanjing 211189, China)

Abstract: To improve the cache utilization in named data networking, a collaborative cache management strategy, named as ACNCM(ant-colony based neighbor cooperation cache management), was proposed. The cache replacement problem in a single node was modeled as a 0/1 knapsack problem firstly, which could be well solved by an ant-colony replacement algorithm. To facilitate the execution of the algorithm, the inner content caching value was calculated by taking into account the size of the content, the access frequency of the content, and the status of neighboring cache copies. Once the to-be replaced content has been figured out, ACNCM was exploited to choose one available collaborative neighbor node to store the content, according to the regular exchange of caching information among neighboring nodes. The experimental results show that ACNCM outperforms the existing caching schemes in terms of cache hit rate, network overhead and the average response latency.

Key words: named data networking, cache management, content caching value, ant-colony replacement algorithm

1 引言

以内容为中心的命名数据网络^[1](named data networking,

NDN)的重要特征之一,是利用网络内置缓存机制提高用户获取内容资源的效率和网络资源的利用率。在 NDN 中,路由节点依靠报文中携带的内容名字转发数据内容,同时

^{*} 国家高技术研究发展计划(“863”计划)基金资助项目(No.2013AA013503),江苏省未来网络前瞻性研究基金资助项目(No.BY2013095-2-06)



每个路由节点都具有缓存本地转发过的数据内容的能力,以便后续相同请求到达时可利用缓存副本直接响应该请求,从而提高响应效率。研究设计高效的缓存管理机制对提升NDN的整体性能具有重要意义。

传统的缓存管理策略有LRU(least recently used)、LFU(least frequently used)、FIFO(first in first out)以及SIZE替换算法等。其中,LRU^[2]将数据内容最近被访问的时间点作为判断依据,在必要时将缓存内容中最近最少访问的数据内容替换出缓存;LFU^[3]统计缓存内容在过去一段时间内被访问的次数(即访问频率),将访问频率最低的内容替换出去;FIFO^[4]缓存管理策略,基于简单的排队规则,根据缓存内容存储的先后顺序依次进行缓存替换;SIZE^[5]替换算法则将数据内容的大小作为缓存替换的主要因素,优先替换缓存中占用空间较大的数据内容。参考文献[6]提出了一种基于内容热度的NDN缓存替换策略。在保留NDN中CS、PIT以及FIB结构的基础上新增了一张CPT(content popularity table),用来保存缓存内容的命中率、历史及当前热度,热度值按计时周期进行加权平均,在缓存替换过程中选择热度最小的内容进行替换。

以上缓存替换策略都是基于单个节点的,并没有将网络当作一个整体来考虑。考虑到NDN中每个路由节点都具有缓存内容的能力,为提高网络整体的缓存价值,相关研究提出了节点间协作的缓存管理策略。根据关注重点、协作决策点的不同,可将节点间协作的缓存管理分为全局协作、路径协作以及邻域协作^[7]。

全局协作根据当前的网络拓扑、节点间的网络距离以及各个节点的访问速率等信息,提前规划好数据内容在网络中的最佳存储位置,使得用户访问内容的“代价”最低。参考文献[8]提出了一种基于内容流行度以及网络拓扑的协作缓存策略,主要思路是根据数据报文距离内容提供端的跳数和数据的流行度,确定报文的生存期以及报文存储的位置,但并没有给出流行度的计算方法以及网络拓扑图的更新方式。

路径协作指数据内容的命中节点到请求节点之间的路由节点,在转发过程中判断是否缓存该内容的协作决策方法。参考文献[9]提出在转发兴趣报文的过程中记录沿途经过的节点信息(如节点状态、请求频率等),命中节点根据请求报文携带的路径上所有节点的状态信息,利用动态线性规划,计算出最优的内容缓存位置。这种方法具有中心处理方式的弊端,在决策节点的计算开销也不小。参考

文献[10]考虑避免复杂的决策算法,在转发内容报文的过程中随机地选择一个节点作为存储内容的路由节点。这种缓存策略易于实现,但未考虑内容的访问热度以及节点的缓存状况。

邻域协作是指在一个节点的邻域范围(经过有限跳数可达的节点,也称邻居节点)内,显式地对缓存放置的位置进行决策,通过协调邻域范围内节点之间的缓存内容,降低网络中数据的冗余度,提高整体的缓存利用率。参考文献[11]提出利用布鲁曼滤波器(bloom filter)技术,根据节点设定的缓存内容时间把内容目录广播给一定区域内的其他节点,设定的保存时间越长,广播的距离越远。该方法没有给出具体的缓存内容时间计算方法,也未通过实验验证其有效性。参考文献[12]提出的邻域协作缓存管理方法主要针对如何降低节点之间的冗余度问题,在后台运行一个降低邻域范围内缓存冗余的程序,定期清理邻域范围内的冗余缓存对象。这种方法可以有效地降低数据的冗余度,但未考虑数据内容本身的信息,会导致某些热门资源在网络存储的副本数量有限,不能很好地满足用户请求。

为充分利用NDN中路由节点的缓存信息,本文针对现有缓存替换策略的不足,基于邻域协作的思想,提出了一种基于蚁群替换算法的邻居协作缓存管理(ACNCM)策略。在单节点的缓存替换中,将自定义的内容缓存价值引入具有较高执行效率和优化求解结果的蚁群替换算法(ACCR)中,提高单个节点的缓存价值;在协作缓存替换中引入协作替换参数,协调不同节点之间的内容缓存价值,从而提高NDN整体缓存价值。

2 基于邻居协作的NDN缓存管理框架

2.1 邻居协作模型与协议交互机制

如图1所示,基于邻居协作的NDN缓存管理框架在现有CS、PIT以及FIB结构的基础上,新添加了邻居信息结构NIB。NIB的主要功能有:在节点计算缓存内容价值时提供邻居副本信息,支持缓存替换算法的运行;在节点执行协作缓存决策过程中提供邻居节点信息,作为判断是否执行协作缓存的一个主要因素;在处理用户发起的兴趣报文过程中,可以优先获取邻居节点中存储的内容,提高响应效率。

邻居信息结构NIB通过邻居缓存信息表(NCT)(见表1)和邻居缓存状态表(NST)(见表2)记录给定邻居深度范围内的所有邻居节点的缓存状态信息,其中邻居深度表

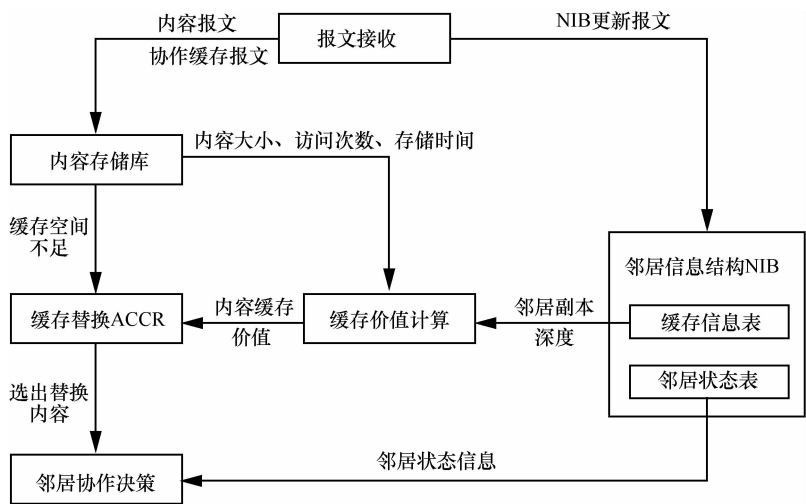


图 1 基于邻居协作的 NDN 缓存管理框架

示该条信息的发送者距接收者的路由跳数，初始值为 1，在转发过程中逐跳增加。

表 1 NCT

名字	邻居深度	邻居节点标识	端口
/test1	2	R ₀	0

表 2 NST

邻居节点标识	邻居深度	缓存空间是否已满	平均内容缓存价值	时延/ms	端口
R ₂	1	否	5.5	26	2
R ₀	2	是	4.7	43	0

采用 ACNCM 策略的节点在接收到兴趣报文时，先检查 CS 和 PIT 中是否有相应记录，若有则按 NDN 标准流程处理；如果都没有，则检查 NIB 以判断邻居节点中是否已有相应的数据内容。如果 NIB 中有相应记录，则将兴趣报文转发给该邻居节点，否则检查 FIB，将兴趣报文通过相应端口转发给上游节点。

为便于路由节点间进行协同缓存，充分利用新增邻居缓存结构中的信息，设计了两种新的报文：NIB 更新报文和协作缓存报文。

(1)NIB 更新报文

NIB 更新报文用于路由节点向其他节点通告本地缓存信息，其结构如图 2 所示。其中，节点标识是节点的唯一性身份标识；最大转发跳数用于限定 NIB 报文在网络中的传输范围；本地缓存是否已满与平均内容缓存价值是判断是否执行协作缓存的两个重要指标，供报文接收端在执行协作缓存策略时采用；缓存内容名字列表记录本地缓存内

容的名字，用于告知邻居节点本地缓存空间存储了哪些内容，供邻居节点执行缓存替换算法以及转发兴趣报文时参考。每个路由节点会根据给定的协作邻居深度以及更新间隔时间定期向外发出 NIB 更新报文。当路由节点接收到 NIB 更新报文时，根据报文中携带的信息更新 NST 中对应的表项；并判断最大转发跳数是否大于 1，若是则将最大转发跳数减 1，并通过当前节点的所有端口转发，否则丢弃该报文。

节点标识	最大转发跳数	报文创建时间	邻居深度
本地缓存是否已满	平均内容缓存价值	缓存内容名字列表	

图 2 NIB 更新报文结构

(2)协作缓存报文

路由节点选出需要替换出缓存的数据内容后，由协作缓存管理策略决定是否对该内容执行协作缓存，如果需要则向选中的邻居协作节点发送包含该内容的报文，称为协作缓存报文，其格式如图 3 所示，其中协作节点标识即协作缓存节点的身份标识，协作邻居深度代表节点选中的协作邻居节点距离本节点的跳数，内容数据即需要协作存储的内容本身。邻居节点收到协作缓存报文后，会首先根据内容缓存价值判断是否应该存储收到的内容，然后根据判断结果处理报文。如果该处理引发了协作邻居节点执行缓存替换，则不再对替换出的数据内容继续执行协作缓存处理，而是直接删除新替换出的内容，以降低不必要的网络开销。



协作节点标识	协作邻居深度	内容数据
--------	--------	------

图3 协作缓存报文格式

2.2 ACNCM 缓存管理的基本思想

ACNCM 缓存管理主要包括两部分,具体介绍如下。

(1) 单节点的缓存替换机制

将缓存替换问题建模为 0/1 背包问题 (knapsack problem, KP), 在单个节点内需要进行缓存替换时, 利用基于蚁群优化算法的缓存替换机制选择需要替换出去的数据内容。在蚁群算法计算选择缓存内容的概率时综合考虑以下因素: 信息素浓度、数据到达的时间、访问次数和数据的大小以及邻居节点中是否存储有相同内容的副本。

(2) 邻域协作缓存管理机制

在协作缓存管理中引入协作替换参数, 对于单个节点中将要被替换出去的数据内容, 计算出各个邻居节点的协作替换参数, 然后选择具有最小协作替换参数的邻居节点作为该数据内容的存储节点。在计算协作替换参数时, 主要考虑邻居节点的平均缓存价值以及访问时延。

3 ACNCM 缓存替换与邻居协作算法

3.1 单节点缓存替换问题建模

给定一个缓存空间有限的网络节点, 当剩余空间不足以存储新到来的数据时, 需要从现有缓存中替换出一些价值相对低的数据内容, 为新到达的数据腾出空间。该问题本质上等同于传统的 0/1 背包问题, 即在给定的约束条件下, 求最大可能组合的优化问题。这是一类典型的 NP 完全问题, 其一般描述为: 给定一个能承受最大重量为 W 的背包和 n 个物品, 物品 i 的重量用 w_i 来表示, v_i 表示物品 i 的价值, 如何选择总价值尽可能高的物品放入背包中, 并且物品的总重量不能超过背包的容量。

基于此认识, 缓存替换问题可描述为: 节点缓存空间最大容量为 S , 当前存储了 n 个数据内容, 数据内容 i 的大小记为 s_i , 对应的内容缓存价值记为 v_i , 待存入缓存的新到达数据内容的大小为 s_n 。需要找出一种存储方案, 使得在数据内容总大小不超过最大可用缓存空间 S 的情况下, 缓存空间中保留的数据内容的缓存价值总和最大。

其形式化描述: 设 $S > 0, 0 < s_i < S, v_i > 0, 1 \leq i \leq n$, 要求解出一个 n 元 0-1 向量 $\{x_1, x_2, x_3, \dots, x_n\}, x_i \in \{0, 1\}, 1 \leq i \leq n$, 满足条件 $\sum s_i x_i \leq S$, 且使得 $\sum v_i x_i$ 最大化。其中 $x_i = 1$ 表示对应的数据 s_i 保存在缓存中, $x_i = 0$ 则表示对应的数据被替换出

缓存。

$$\text{数学模型: } \max \sum_{i=1}^n v_i x_i, \text{ 约束条件: } \sum_{i=1}^n s_i x_i \leq S。$$

3.2 数据内容的缓存价值计算

数据内容的缓存价值与背包问题中的物品价值相对应, 是对节点缓存空间中存储的数据内容的价值估量。由于网络中数据内容请求的随机性, 无法直接确定数据内容的缓存价值, 只能根据数据内容在缓存中以往的效用来判断内容的价值。本文定义内容 i 的缓存价值如式(1)所示。

$$Val_i(t) = \frac{1}{s_i} \times \frac{hitcount_i}{t-t_i} \times (1 + neighbordeep_i) \quad (1)$$

其中, s_i 表示内容 i 的体积, 也即占用的缓存空间, 数据内容的体积越大, 其缓存价值越小。 t 表示当前时间, t_i 表示内容被存储的时间, $hitcount_i$ 表示内容在时间段 $(t-t_i)$ 内被访问的次数, $hitcount_i/(t-t_i)$ 表示内容在该时间段内的访问频率, 它反映数据内容在过去一段时间的热度, 故缓存价值与其成正比。 $neighbordeep_i$ 表示本节点到存储内容 i 副本的邻居节点的跳数, 如果没有邻居节点存储该内容, 则其值为网络设定的最大协作邻居深度值加 1。邻居深度越大, 从邻居节点获取数据内容的开销越大, 在本节点就更有存储的价值, 所以邻居深度与缓存价值成正比。

3.3 基于蚁群算法的单节点缓存替换机制

求解 0/1 背包问题有很多种方法, 包括图论法、动态规划法、分支限界法等精确求解方法以及蚁群算法、遗传算法、模拟退火算法等启发式求解方法。在这些算法中, 蚁群算法具有结果精确度高、计算时间短等优点, 参考文献[13]给出了一种求解 0/1 背包问题的快速蚁群算法 (KPACA)。考虑到 NDN 缓存对象的属性信息满足蚁群算法的计算要求, 本文借鉴蚁群算法的主要思想, 提出一种新的基于蚁群算法的缓存替换机制 (ACCR)。

3.3.1 蚁群信息素的计算

在蚁群算法中, 信息素是蚂蚁群体选择和演化过程的一种体现, 信息素随着选择的蚂蚁数量的增多而增加, 同时也会随着时间的增长而逐渐挥发。但与求解传统 TSP 旅行商问题的蚁群算法不同, 在求解 0/1 背包问题的蚁群算法模型中, 信息素积累在蚂蚁选过的物品上而不是走过的路径上。本文参照 MMAS 算法^[14]中信息素的更新方式, 在执行完一轮选择后仅对蚂蚁找出的最优解中包含的数据内容进行信息素的增加修改, 其余没有被最优解选中的数据内容信息素相应地减少。蚁群信息素 $\tau_i(t)$ 的更新如式(2)

所示。

$$\tau_i(t+p)=(1-\rho)\tau_i(t)+\Delta\tau_i \quad (2)$$

其中, $\rho \in (0,1)$ 为信息素的挥发系数, $(1-\rho)$ 表示信息素的残余因子; $\Delta\tau_i$ 表示在时间段 p 内信息素的增量, 初始值为 0。信息素增量 $\Delta\tau_i$ 如式(3)所示。

$$\Delta\tau_i = \begin{cases} \rho\tau_i(t)+0.1, & \text{内容 } i \text{ 在本轮最优解中} \\ -\frac{s_i}{S}\rho\tau_i(t), & \text{其他} \end{cases} \quad (3)$$

式(3)中考虑到体积大的内容占用的缓存空间也大, 当内容没有被蚂蚁最优解选中时, 会根据内容的大小增加它的挥发程度。

3.3.2 选择概率的计算

在蚁群算法中, 每个物品都有它被蚂蚁选中的概率, 称为选择概率, 记为 P_i^k 。参考 KPACA^[13], 本文中蚂蚁 k 在一轮执行过程中选择缓存对象 i 的概率 $P_i^k(t)$ 如式(4)所示。

$$P_i^k(t) = \begin{cases} \frac{\tau_i^\alpha(t)\eta_i^\beta(t)}{\sum_{i \notin \text{pitched}_k} \tau_j^\alpha(t)\eta_j^\beta(t)}, & i \notin \text{pitched}_k \\ 0, & \text{其他} \end{cases} \quad (4)$$

其中, $\tau_i(t)$ 为物品 i 上的蚁群信息素, $\eta_i(t)$ 为启发式函数; α, β 为信息素因子和期望启发式因子, 分别反映蚂蚁在选择内容时信息素和缓存内容属性对选择结果的影响, α 值越大, 表示蚂蚁更倾向于依赖之前蚂蚁的选择结果; β 值越大, 则表示蚂蚁独自判断选择的能力越大。 pitched_k 表示蚂蚁已经判断过, 不能再选择的内容的集合。

启发式函数 $\eta_i(t)$ 表示蚂蚁选择缓存对象 i 的期望程度, 启发式函数根据数据内容自身的属性决定蚂蚁对缓存内容的选择概率。在选择概率中加入启发式函数, 一方面可以帮助蚂蚁加快选择出最优解的速度, 另一方面也是结合数据内容的实际使用情况, 将缓存内容自身的信息作为选择判断的重要因素。参考文献[13]中把启发式函数定义为物品单位体积的价值, 认为物品单位体积的价值越高, 选择该物品的期望就越高。由于本文在内容缓存价值计算中已经考虑了内容体积大小对缓存价值的影响, 所以选择内容缓存价值作为期望启发式信息, 即 $\eta_i(t) = \text{Val}_i(t)$ 。因此, 缓存内容价值越大, 留在缓存空间的期望就越大; 相反, 缓存价值低的数据内容应尽早替换出去。

3.3.3 基于蚁群算法的缓存替换机制

当路由节点的缓存空间不足以存储新内容时, 根据 ACCR 算法进行内容替换, 使得节点中缓存的数据保持较

高的内容价值。ACCR 过程描述如下。

(1) 当节点收到存储内容请求时, 根据报文内容创建并初始化缓存数据对象。访问命中次数 hitcount_i 初始化为 1, 存储时间记录为当前时间。

(2) 计算剩余缓存空间是否能够存储该数据内容。如果空闲缓存空间能够容纳该内容, 则直接存储内容, 结束; 否则, 转步骤(3)。

(3) 初始化蚁群替换算法参数, 设各内容对象的信息素 $\tau_i(t)$ 为一固定值 τ_0 。设定算法参数 α, β 和 ρ , 设定时间 t 为系统当前时间。设定最大迭代次数 $\text{loop}_{\max}$, 模拟蚂蚁的个数 m_a 。初始化每个蚂蚁的禁忌表 pitched_k 和解集 result_k 。

(4) 让每只蚂蚁 k 根据式(4)计算出缓存内容被选择的概率 $P_i^k(t)$, 蚂蚁 k 根据该概率值选择下一个缓存内容 i , 并将该内容加入禁忌表 pitched_k 中, 表示该内容已经被选择过。然后试着将该内容放入解集 result_k 中, 如果该解集的内容总体积小于节点最大缓存容量 S , 则把该内容并入解集 result_k 中作为可行解的一部分。循环上述过程, 直到 m_a 只蚂蚁都做完了对所有缓存的选择判断, 并构造出各自的解集。

(5) 找出本轮循环结束后所有蚂蚁的解集中缓存价值最大的解集 $\text{result}_{\max}$, 如果该解集的总缓存价值大于当前最优解, 则替换当前最优解。如果迭代次数小于最大迭代次数并且不满足结束条件, 则更新信息素并转至步骤(4); 否则, 转至步骤(6)。

(6) 如果新到达的数据内容在最优解中, 则将新到达内容存入本地缓存, 不在最优解中的缓存内容替换出去; 否则, 不存储新到来的数据内容。缓存替换执行结束。

3.4 基于蚁群替换算法的邻居协作缓存管理策略

缓存替换算法找出需要替换的内容之后, 启用协作缓存机制查询邻居节点是否存储了该内容的副本, 若是, 则直接删除该内容不做协作缓存处理; 否则, 根据邻居节点的缓存状态信息, 计算协作替换参数, 选择合适的协作缓存节点。节点 e 的邻居节点 j 的协作替换参数 ρ_e^j 计算如下:

$$\rho_e^j = \frac{1}{\text{Val}_{\text{avg}}^j} \times \frac{1}{\text{delay}^j} \quad (5)$$

其中, delay^j 为对邻居节点 j 的访问时延, $\text{Val}_{\text{avg}}^j$ 为邻居节点 j 的平均内容缓存价值。由于两个相邻节点之间的通信状况不好, 会导致将来从该邻居节点获取缓存数据时花费较高的时间成本, 所以在选择缓存协作节点时, 考虑了邻居节点之间的通信时延。此外, 由于平均内容缓存价值



可以反映一个节点存储的数据内容的整体质量,因此应选择平均缓存价值较低的邻居节点作为协作节点,以提高邻居节点的整体缓存价值。

节点协作缓存模块的处理流程如图4所示。

4 实验仿真与性能分析

4.1 实验环境构建

为验证本文提出的缓存管理策略的可行性和有效性,对 ACNCM 策略进行了模拟实验。采用 C++ 及 MATLAB 语言模拟以事件为驱动的协作缓存管理系统。

首先模拟实验过程中的数据输入,依据现实生活中用户对互联网内容资源的访问特征,实验中设定用户发起的请求序列符合柏拉图定律,即 80% 的请求报文对应于 20% 的内容。用户对内容请求的时间间隔服从指数分布。然后实现 ACNCM 的逻辑处理部分,这是整个实验的核心部分,用来模拟缓存替换算法以及基于邻居协作的缓存管理过程。

网络拓扑规模根据 GT-ITM 拓扑生成工具生成一个节点数为 10 的随机网络拓扑图,其中任意两个节点之间存在直达路径的概率为 0.4。从随机生成的拓扑图的边缘节点中选择一个节点作为内容提供节点,选取两个边缘节点作为客户端节点。相邻路由节点之间的时延设为 20 ms, NIB 报文的最大转发跳数设为 5。实验中算法的相关参数 α, β, ρ 的取值分别为 0.4, 0.3 和 0.7, 协作邻居深度设定为 1。模拟实验的具体参数配置见表 3。

表 3 实验参数

配置	参数
实验时间	60 min
节点数量	10 个
内容总数	1 000 条
每秒平均发送请求报文数量	2 条
邻居信息更新间隔时间	15 s

4.2 缓存替换算法 ACCR 的性能分析

首先考察单节点缓存替换算法 ACCR 的性能。在相同

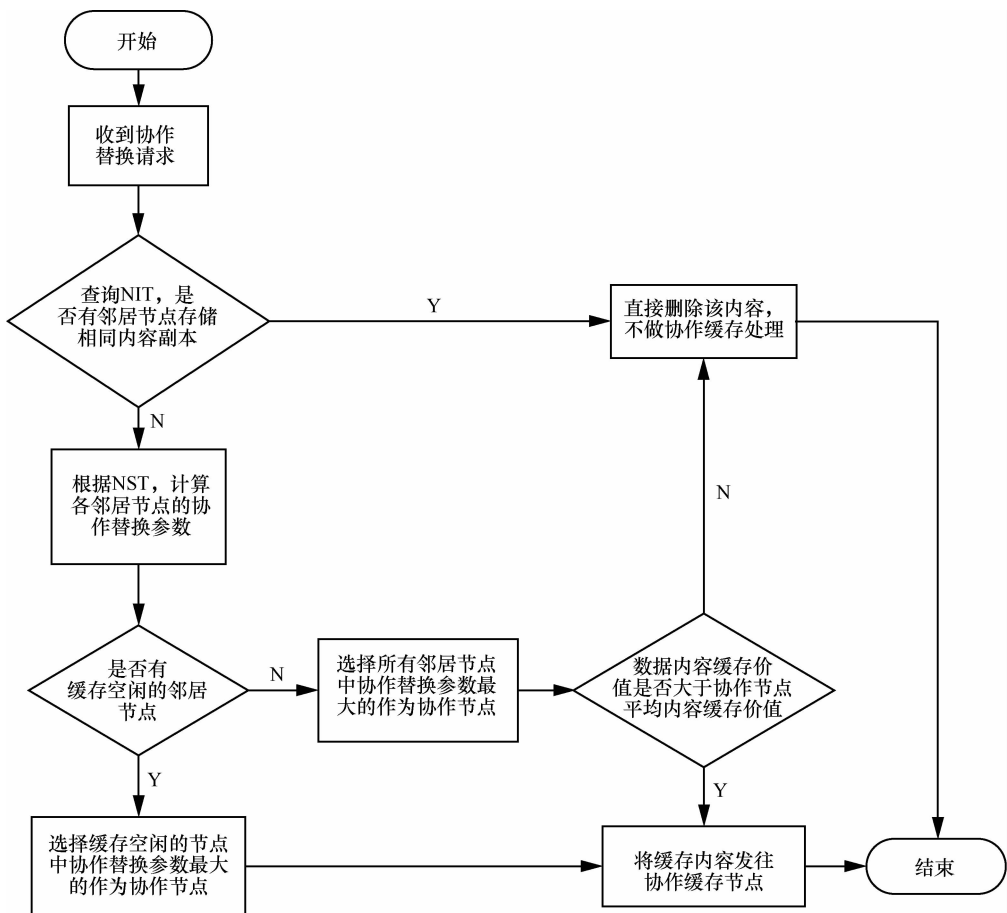


图 4 协作缓存流程

的实验环境下,针对路由节点的缓存容量为 50 MB 和 80 MB 两种情况,分别进行 ACCR 算法、LRU 算法和 FIFO 算法的实验,运行结果如图 5、图 6 所示。

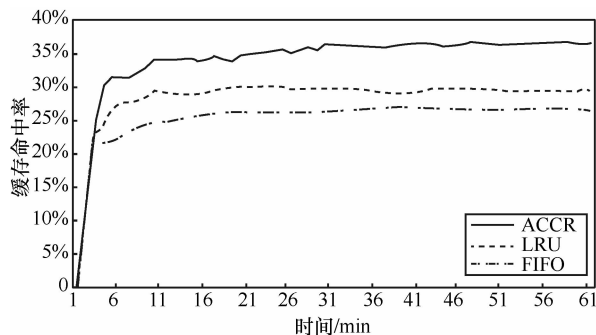


图 5 缓存空间为 50 MB 时的缓存命中率

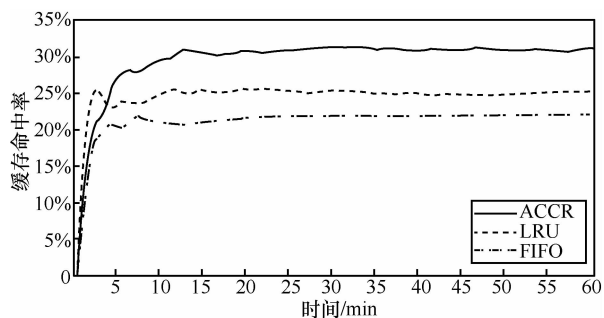


图 6 缓存空间为 80 MB 时的缓存命中率

由图 5 和图 6 可以看出,ACCR 的缓存命中率高于 LRU 算法和 FIFO 算法。分析其原因,ACCR 在执行缓存替换时综合考虑了缓存对象自身的属性信息,如缓存数据的大小、访问次数、邻居副本因素、存储时间等,同时结合蚁群信息素的正反馈机制,使得缓存中保留了那些缓存价值较高的数据内容。

4.3 协作缓存策略 ACNCM 的性能分析

引入邻居协作缓存机制,考察在相同的实验环境下, FIFO、LRU、ACCR 和 ACNCM 4 种缓存管理策略的报文开销和平均时延。报文开销定义为网络报文总数与用户得到的响应报文的比值。如图 7 所示, FIFO 缓存管理策略产生的报文开销最大, LRU 缓存管理策略次之; ACCR 的效果比 LRU 好,但是相比于 ACNCM 性能又相对较差。说明 ACNCM 能有效降低全网络的报文数量,降低网络总开销。

如图 8 所示的实验结果表明, ACNCM 的平均时延要优于其他缓存管理策略,这主要是由于邻居协作缓存管理策略充分利用了邻居节点的缓存功能。通过协作存储内容使得邻居节点之间作为一个整体,存储了缓存价值较高的

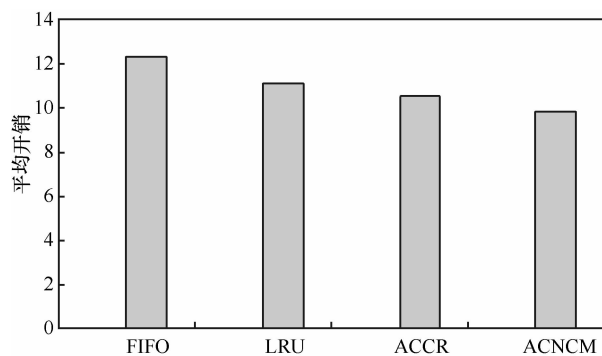


图 7 报文开销比较

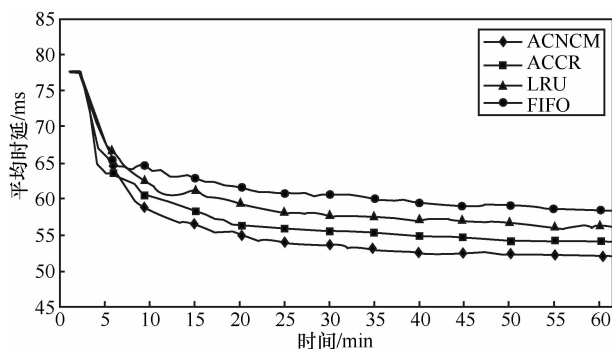


图 8 平均时延比较

缓存内容,更高效地利用了整体的缓存空间。当用户请求到达时, ACNCM 可以更快地响应用户请求,提高用户对内容资源请求的响应效率,降低了响应时延。

5 结束语

针对 NDN 缓存利用效率问题, 本文根据数据内容自身的属性信息以及邻居节点的存储状态信息, 提出一种基于邻居协作的缓存管理 (ACNCM) 策略。本文的主要工作包括: 定义了缓存数据的内容价值属性, 结合邻居节点的缓存信息, 为节点缓存替换决策提供重要参考; 将缓存替换问题建模为 0/1 背包问题, 利用蚁群优化算法, 结合蚁群信息素和缓存内容自身的属性信息, 对节点的缓存空间进行替换管理; 采用邻域协作思想, 依据内容缓存价值和邻居缓存信息, 选择合适的邻居节点作为协作缓存节点, 提升邻居节点间整体缓存利用率。

实验结果表明, ACNCM 机制在缓存的命中率、用户请求的响应时延以及网络整体开销等方面, 相比现有缓存管理方法都有不同程度的优化。进一步的研究工作包括探讨与协作替换参数相关的邻居属性, 优化 NIB 更新报文, 实现更准确的节点间协作开销估算。此外, 增加更多同类算法的对比也是未来要完成的工作之一。



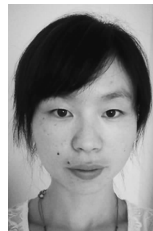
参考文献

- 1 Zhang L, Estrin D, Burke J, *et al.* Named Data Networking (NDN) Project. Technical Report NDN-0001, Xerox Palo Alto Research Center-PARC, 2010
- 2 Willick D L, Eager D L, Bunt R B. Disk cache replacement policies for network filesystems. Proceedings of the Distributed Computing Systems, California, USA, 1993: 2~11
- 3 张震波, 杨鹤标, 马振华. 基于 LRU 算法的 Web 系统缓存机制. 计算机工程, 2006, 32(19): 68~70
- 4 Chrobak M, Noga J. LRU is better than FIFO. Proceedings of the Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, San Francisco, California, USA, 1998: 78~81
- 5 Abrams M, Standridge C R, Abdulla G, *et al.* Removal policies in network caches for world-wide web documents. Proceedings of the ACM SIGCOMM, Stanford, CA, USA, 1996: 293~305
- 6 李韬, 李玉宏. 一种基于内容热度的 NDN 缓存替换算法. 中国科技论文在线, 2012
- 7 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
- 8 Ming Z, Xu M, Wang D. Age-based cooperative caching in information-centric networks. Proceedings of the IEEE INFOCOM WKSHPs, Orlando, FL, USA, 2012: 268~273
- 9 刘外喜, 余顺争, 蔡君等. ICN 中的一种协作缓存机制. 软件学报, 2013, 24(8): 1947~1962
- 10 Eum S, Nakauchi K, Murata M, *et al.* CATT: potential based routing with content caching for ICN. Proceedings of the Second Edition of the ICN Workshop on Information-Centric Networking, Istanbul, Turkey, 2012: 49~54
- 11 Wang Y, Lee K, Venkataraman B, *et al.* Advertising cached

contents in the control plane: necessity and feasibility. Proceedings of the IEEE INFOCOM WKSHPs, Orlando, FL, USA, 2012: 286~291

- 12 Wang J M, Zhang J, Bensaou B. Intra-AS cooperative caching for content-centric networks. Proceedings of the 3rd ACM SIGCOMM Workshop on Information-Centric Networking, Santa Barbara, CA, 2013: 61~66
- 13 王会莹, 贾瑞玉, 章义刚等. 一种求解 0/1 背包问题的快速蚁群算法. 计算机技术与发展, 2007, 17(1): 104~106
- 14 Stutzle T, Holger H H. Max-min ant system. Future Generation Computer System, 2000, 16(8): 889~914

[作者简介]



董利利,女,东南大学硕士生,主要研究方向为融合环境下的缓存管理。

王勇,男,东南大学硕士生,主要研究方向为内容中心网络缓存管理。

董永强,男,博士,东南大学副研究员,主要研究方向为网络融合技术、移动网络计算。

杨鹏,男,博士,东南大学副教授,主要研究方向为未来网络体系结构、双结构互联网等。

(收稿日期:2014-07-20)

· 简讯 ·

2014 年 7 月通信业主要指标完成情况(二)

指标名称	单位	本月末到达	比上年末净增	本月净增
固定电话用户合计	万户	25 696.1	-1 002.4	-130.6
公用电话用户	万部	2 120.3	-113.1	-10.2
城市电话用户	万户	17 983.7	-473.1	-79.5
住宅电话用户	万户	10 141.2	-333.1	-50.3
农村电话用户	万户	7 712.4	-529.3	-51.1
住宅电话用户	万户	6 130.1	-513.4	-52.3
移动电话用户合计	万户	126 340.6	3 429.3	297.4
其中:3G 用户	万户	47 690.3	7 529.2	613.4
互联网拨号用户	万户	453.3	-31.8	-6.6
互联网宽带接入用户	万户	19 740.7	849.8	-49.2
其中:xDSL 用户	万户	9 984.9	-731.6	-128.0
移动互联网用户	万户	87 220.9	6 464.6	1 103.7
固定电话普及率	部/百人	19.0		
移动电话普及率	部/百人	92.6		

(来源:工业和信息化部运行监测协调局)