

# 基于“Che近似”的CCN缓存资源估算

郑焯<sup>1,2</sup>, 黄泽宇<sup>2</sup>, 徐正欢<sup>3</sup>, 施钱宝<sup>2</sup>

1. 中国科学技术大学自动化系 合肥 230026
2. 合肥综合性国家科学中心人工智能研究院 (中国科学技术大学) 合肥 230088
3. 中国科学技术大学先进技术研究院 合肥 230088

E-mail: [qzheng@ustc.edu.cn](mailto:qzheng@ustc.edu.cn)

**摘要:** 随着网络视频点播, 流媒体等业务的多样化发展, 相同的内容在传输和存储过程中占据了传统网络的大量资源。为了从根源上解决上述问题, 内容中心网络CCN以其网内通用缓存的优势引起业界广泛关注。本文区别于传统的CCN缓存资源分配研究, 从需求出发, 基于“Che近似”层次缓存建模理论及其扩展研究, 提出一种面向接入网的小型树状拓扑缓存资源估算方法。该方法量化了系统平均缓存命中率这一网络规划目标与网络拓扑中节点缓存容量的关系。根据节点的缓存命中率目标估算节点缓存容量大小, 为网络中的节点提供缓存的配置信息。实验仿真表明, 本文的估算方法对于网络运营商规划和部署内容服务网络具有参考意义和价值。

**关键词:** 内容中心网络, 缓存估算, 缓存分配, Che近似

## CCN Cache Resource Estimation Model Based on “Che Approximation”

Zheng Quan<sup>1,2</sup>, Huang Zeyu<sup>3</sup>, Xu Zhenghuan<sup>3</sup>, Shi Qianbao<sup>2</sup>

1. *Department of Automation, University of Science and Technology of China, Hefei 230026*
2. *Institute of Artificial Intelligence (University of Science and Technology of China)*
3. *Institute of Advanced Technology, University of Science and Technology of China, Hefei 230026*

**Abstract:** With the diversified development of online video-on-demand, streaming media and other services, the same content occupies a large amount of resources in traditional networks during transmission and storage. In order to solve the above problems from the root cause, the content-centric network CCN has attracted widespread attention in the industry due to its advantage of general cache in the network. Different from the traditional CCN cache resource allocation research, this paper proposes a small tree topology cache resource estimation model for access networks based on the "Che approximation" hierarchical cache modeling theory and its extended research. The model quantifies the relationship between the average system cache hit rate, a network planning objective, and the node cache capacity in the topology. According to the cache hit rate target of the node, the cache capacity of the node is estimated, and the storage configuration information is provided for the nodes in the network. Experimental simulations show that our estimation model has certain reference significance and value for network operators to plan and deploy content service networks.

**Key Words:** Content Centric Networks, Cache Estimation, Cache allocation, Che Approximation

### 1 文件类型要求

随着网络视频, 流媒体等业务的快速兴起, 未来将有超过80%的全球数据流量来自于视频内容的分发者<sup>[1]</sup>。由于网络中大规模用户访问所产生的数据流量, 尤其是视频业务, 给当前的互联网带来了巨大的流量负担。且视频业务中, 很多用户订阅和点播的内容具有高度的趋同性, 因此网络中充斥着大量重复的、冗余的数据内容。造成网络资源利用率低, 资源浪费的现象。

内容中心网络(CCN)<sup>[2]</sup>的提出, 能够有效地解决上述问题。CCN中节点具有缓存资源, 可以有效地减少网络中的流量负载和节约链路带宽。对于CCN网络中的每一个节点都可以看作是数据流量的低通滤波器, 请求频率较高的内容请求, 可以缓存在节点的缓存队列中, 并对后续到来的相应的请求进行响应。而对于请求频率较低的内容请求, 节点缓存队列中缓存该内容的概率很低, 大部分这样的请求会向上转发。CCN中的节点的缓存容量的大小决定了向上转发的

\*2020年安徽省重点研发计划(202004a05020078); 国家网络创新基础设施项目: 国家未来智能网络试验设施(2016-000052-73-01-000515);

频率, 对于一个网络拓扑来说, 缓存空间的大小影响着链路中的流量。

CCN的网内缓存需要在路由器上分配存储资源, 而这些资源往往是极其昂贵和耗能的。因此如何估算网络中节点的缓存容量, 合理地规划缓存资源, 提高网络中节省的流量转发带来的收益以及缓存为用户带来的体验, 是网络运营商们目前关注的焦点。

目前 CCN 缓存资源的相关研究, 集中于对缓存资源分配的探索, 该类研究大都遵循 Jacobson 等人<sup>[3]</sup>提出的假设, 认为缓存是由配备了内容存储的路由器提供的。由于受到技术和成本的限制, 其容量是有限的。因此相关研究内容多采用各种优化目标或者协作性缓存策略来最佳的利用总体存储容量。文献[4]中作者定义了一个新的度量指标-请求影响度(RID), 并基于 RID 进行内容存储空间的网络异构分配方案。文献[5]中作者提出了以最小化用户能量消耗为目标的缓存资源分配算法, 将缓存资源优化分配给距离更近的用户, 形成一个时延和能量消耗最小的网络。文献[6]中作者提出节点缓存命中方差最小化和请求延迟最小化为目标的缓存决策模型, 通过一种带阈值的启发式算法来合理分配资源。

这些研究主要出发点都是从充分利用缓存空间的角度出发, 前提是总缓存空间, 或者单个节点的缓存空间大小是已知的, 或先验固定的, 通过提出或优化各种内容的放置策略以及替换策略, 尽可能的提升节点的缓存命中率。

而CCN网络部署之前, 需要根据目标按需对网络资源进行估算, 因此与缓存性能优化研究不同的是, 缓存的容量是未知的, 目前还没有面向内容服务网络背景下关于缓存资源估算相关的研究。因此综合上述对当前 CCN 中缓存资源的研究现状概述以及现有技术的可行性分析, 本文基于“Che近似”<sup>[7]</sup>理论及其扩展研究<sup>[8,9]</sup>, 提出了一个两层树状拓扑节点缓存资源估算方法。该方法描述了系统的平均缓存命中率与单节点缓存命中率之间的关系, 根据目标缓存命中率, 估算网络中各个节点所对应的缓存容量大小。通过ndnSIM进行实验仿真, 观察估算方法的准确性。实验结果表明, 本文的估算方法可以为CCN网络的前期部署和规划提供参考。

本文结构组织如下: 首先概述本文工作动机与主要工作; 第2节介绍缓存分配的相关研究工作; 第3节给出两层树状拓扑的缓存资源估算方法; 第4节给出了估算方法的详细分析; 第5节进行仿真实验, 给出实验结果; 第6节总结本文工作。

## 2 相关工作

### 2.1 网络资源建模研究

Che等人<sup>[7]</sup>针对独立参考模型(IRM)下的LRU替换策略提出了“Che近似”理论, 它的主要思想在于提出常数特征时间的概念, 该特征时间表示内容在缓存中的最大生存时间, 即内容自前次请求后从缓存中逐出的时间, 解耦了内容之间的关联性, 并给出单个节

点和树形拓扑根节点处内容向上层节点(服务器)传输速率的数学解析式, 简化了缓存系统的分析工作。该模型可以用于分析缓存建模问题。

Martina等人<sup>[8]</sup>通过对“Che近似”的扩展和推广, 提出了一种统一的方法来分析缓存的性能, 该方法分析了几种缓存策略(LCE、LCP和LCD)和替换策略下(LRU、q-LRU、FIFO、RANDOM等)的缓存性能, 不同于“Che近似”局限于IRM流量下的LRU缓存, 该方法具有更广泛的适用性, 可以作为缓存系统的一般性能评估的基础。

Rizk等人<sup>[9]</sup>在Che等人<sup>[2]</sup>和Martina等人<sup>[3]</sup>的工作基础上, 提出丢失率的概念, 通过应用迭代方法, 利用丢失率来计算系统命中率, 在缓存层次结构中引入了一个新的命中率计算模型。该方法对缓存层次结构的命中率和其他指标建模的效果明显优于以前的工作。

### 2.2 网络资源分配研究

网络存储是互联网体系架构中重要的资源。优化资源的分配和部署结构, 提升网络资源的利用率, 是网络技术发展持续关注的问题。

当前关于 CCN 网络中缓存资源分配研究大多遵循 Jacobson 等人<sup>[3]</sup>提出的缓存是由配备了内容存储的路由器提供的。由于物理上的限制, 其缓存容量是有限的。因此在这些研究中, 主要出发点是从充分利用缓存空间的角度出发, 前提是总缓存空间, 或者单个节点的缓存空间大小是已知的, 或先验固定的, 通过提出或优化各种内容的放置策略以及替换策略, 尽可能的提升节点的缓存命中率。

文献[10-12]中关于 CCN 缓存性能的研究均为同构缓存分配方式, 即网络中节点的缓存空间具有相同的容量。该部署方式配置简单, 用于性能分析时, 复杂度低。但该分配方式无法按需部署, 相同的缓存容量不能很好的服务于不同的用户需求, 缓存资源利用率较低。实际网络部署过程中, 会有大量不同容量规模的路由器可供选择, 而不可能是同等规模的路由器。因此需要为每个路由器分配适当的缓存容量, 以提高网络性能, 并减少经济投资。

Rossi 等人<sup>[13]</sup>提出在网络的核心路由器中部署更多的缓存空间, 核心路由器按照图拓扑节点度数、紧密性、图中心性和介数等中心性度量指标来定义, 根据路由器的中心性值按比例分配缓存容量, 该异构部署方式可以灵活的根据网络拓扑的结构, 调整缓存节点空间的大小。但结论是缓存容量的异构性部署带来的增益有限, 与其考虑各种节点重要性的度量指标, 按照节点的度这种最简单方式进行差异化缓存空间分配, 整体上就具有很好的系统鲁棒性。

Cui 等人<sup>[4]</sup>认为, CCN 网络中以内容为中心的数据传输模式与传统网络不同, 只根据拓扑图的结构属性无法反映网络中流量的分布特征以及用户的请求行为特征。节点的重要性要考虑内容的请求数量和节点的服务速率, 并定义了请求影响因子 RIF (Request

Impact Factor) 和请求影响度 RID (Request Influence Degree), 来计算节点的重要性程度。该方案与同构分配, 以及按照拓扑结构的度属性分配缓存容量的方式相比, 可以更好根据网络节点的实际负载情况, 较合理的按需分配。但该方法也只是面向CCN网络进行优化的启发式算法, 网络中节点的服务速率(即节点的平均请求命中率)受节点的缓存空间大小, 内容流行度变化等多种因素影响。因此该方法无法得到最优的资源分配方案。

综合上述从单方面属性对缓存容量进行同构和异构分配的研究, Wang 等人<sup>[14,15]</sup>提出了一种用于优化缓存容量分配的 OPT 算法, 将缓存容量的分配问题转化为标准的背包问题, 来最大化流量收益。作者论证了该方法采用计算推导的方式获取缓存空间的分配方案, 相比于启发式算法, 可以更精确的计算出节点缓存空间分配的容量大小, 同时在减轻服务器负载上性能表现更优, 但缺点是当网络拓扑规模增大时, 算法的执行时间成指数级别增长。

无论是采用启发式算法, 还是通过建模, 将问题转化成最优化问题的算法, 研究的重点都集中在网络缓存资源的分配上, 即如何在网络总存储预算有限的情况下, 合理分配缓存容量, 提高缓存性能。

而实际上, 运营商在构建和部署网络时, 往往需要根据当前用户或接入网的规模, 按照用户需求或要达到的预期性能指标, 估算网络的资源使用量。因此, 在节点缓存容量未知的前提下, 设计一种从性能指标量化到所需资源的 CCN 缓存估算方法, 更适用于 CCN 网络的实际部署和规划场景的研究。

### 3 系统模型

节点可以根据到服务器的距离分为不同的层次<sup>[6]</sup>, 在本节中提出了一种两层树状拓扑缓存估算方法, 作为小规模接入网络拓扑的抽象。该估算方法的节点缓存容量是未知的, 需要根据需求或性能指标进行估算。该方法基于“Che近似”理论及其扩展研究, 根据目标命中率, 分析缓存节点的特征时间, 对节点的缓存容量进行估算分析。

#### 3.1 基本框架

为了简化层次网络拓扑, 如图 1 所示, 从图中可以看出, 网络的建模包含第一层缓存节点、第二层缓存节点和一个内容服务器。节点的缓存决策策略使用 LCE(level copy everywhere), 缓存替换策略使用 q-LRU。q-LRU 与传统的 LRU 主要区别在于: 当请求到达时, 若缓存中没有相应内容, 则返回的数据以概率 q 插入缓存, 这样可以有效地降低流行度低的内容的缓存概率, 提高流行内容的命中率。用户通过第一层缓存节点接入网络, 内容的请求先到达第一层缓存, 若命中则返回数据, 否则向上转发; 返回的相应的数据, 以 q 的概率缓存下游节点上。假设用户的请求速率为  $\lambda$ , 请求内容的流行度服从 Zipf 分布。

该方法设定预期的目标命中率, 对节点的缓存容量进行估算。首先设定第一层命中率, 根据“Che近似”理论及其扩展研究, 计算出第一层各节点的特征时间, 进而估算出第一层每个节点的缓存容量。对于第二层节点的缓存容量, 先根据系统的目标命中率和第一层的等价命中率, 求出第二层节点的平均命中率, 并根据第二层请求的到达情况, 估算出第二层节点的特征时间, 计算出第二层节点的缓存容量。

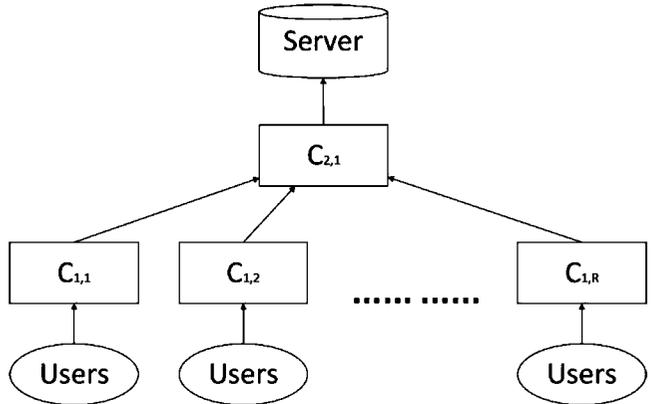


图 1: 估算方法网络拓扑

#### 3.2 “Che近似”

“Che近似”针对独立参考模型 (IRM) 下的 LRU 替换策略, 它的主要思想在于提出常数特征时间  $T_C$  的概念, 其中  $T_C$  表示内容在大小为  $C$  的缓存中的最大生存时间, 即内容  $i$  自前次请求后从缓存中逐出的时间, 解耦了内容之间的关联性, 该模型可以用于分析缓存建模问题。

对于  $N$  个不同的内容到达大小为  $C$  的缓存, 到达率  $\lambda_i, i \in \{1, \dots, N\}$  服从泊松到达过程, 内容存在于缓存中的概率为  $p_{in}(i) = 1 - e^{-\lambda_i T_C}$ 。对于泊松到达, 内容  $i$  命中的概率即意味着在缓存中找到  $i$  的概率, 因此对于第一层节点来说, 内容  $i$  的命中率等于其存在率。从期望的角度将内容存在于缓存中的概率与缓存容量  $C$  联系起来, 可以得到:

$$\sum_{i=1}^N p_{in}(i) = \sum_{i=1}^N (1 - e^{-\lambda_i T_C}) = C \quad (1)$$

对于请求过程服从泊松分布, 内容  $i$  的初始流行度  $\hat{\lambda}_i$  满足 Zipf 分布, 第一层缓存的平均命中率, 只需对内容命中概率进行相应的加权, 即:

$$p_1 = \sum_{i=1}^N \hat{\lambda}_i p_{hit}(i) \quad (2)$$

其中  $\hat{\lambda}_i = \frac{K}{R(i)^\alpha}$  为内容流行度,  $K = \frac{1}{\sum_{i=1}^N \frac{1}{R(i)^\alpha}}$  为归一化因子,  $R(i)$  为内容  $i$  的流行度排名,  $\alpha$  为 Zipf 参数。并且有  $\lambda_i = \hat{\lambda}_i \sum_{i=1}^N \lambda_i$ , 并且  $\sum_{i=1}^N \lambda_i = \lambda$ ,  $\lambda$  为第 1 层节点的内容请求速率。

使用“Che近似”理论来估算缓存容量, 需要预先设置目标命中率, 根据目标命中率求解不动点方程式 (2), 计算出节点的特征时间, 再根据特征时间算出

每个内容的存在概率,最后根据式(1)计算从节点的缓存容量。

本文中使用的符号含义如表1所示。

表1. 符号说明

$C_{1,r}$	第一层节点 r 的缓存容量
$N$	网络中内容的数量
$\tilde{p}_l$	第 l 层的平均命中率
$\tilde{p}_{sys}$	系统命中率
$p_{hit}(i, 1, r)$	内容 i 在第 1 层节点 r 上的命中率
$p_{hit}(i, l)$	内容 i 在第 l 层的命中率
$p_{in}(i, 1, r)$	内容 i 在第 1 层节点 r 上的存在概率
$p_{in}(i, l)$	内容 i 在第 l 层的存在概率
$R$	第一层节点的数量
$m_{i,l,r}$	内容 i 在第 l 层节点 r 上的丢失率
$m_{i,l}$	内容 i 在第 l 层的平均丢失率
$\hat{\lambda}_i$	内容 i 的流行度/请求概率
$\lambda_i$	内容 i 在第一层的到达率
$\bar{\lambda}_i$	内容 i 在第二层的到达率

## 4 模型分析

### 4.1 第一层节点缓存容量分析

从[8]出发,本文分析了 $q \in (0,1]$ 的q-LRU缓存。对于第一层缓存节点 r 来说,在 t 时刻内容 i 存在于 r 的缓存中,可以得出相应的结论:对于上一次的请求到达时刻 $\tau \in (t - T_{1,r}, t)$ ,要么内容 i 在时刻 $\tau^-$ 已存在于缓存中,要么它是以 q 的概率插入缓存。将内容 i 存在于缓存中的概率记为 $p_{in}(i, 1, r)$ ,有:

$$p_{in}(i, 1, r) = (1 - e^{-\lambda_i T_{1,r}})[p_{in}(i, 1, r) + q(1 - p_{in}(i, 1, r))] \quad (3)$$

根据上式,可以得到内容 i 存在于 r 的缓存中的概率 $p_{in}(i, 1, r)$ 为:

$$p_{hit}(i, 1, r) = p_{in}(i, 1, r) = \frac{(1 - e^{-\lambda_i T_{1,r}})q}{1 - (1 - e^{-\lambda_i T_{1,r}})(1 - q)} \quad (4)$$

将公式(4)代入(2)可以得到节点 r 的命中率为:

$$p_{1,r} = \sum_{i=1}^N \hat{\lambda}_i \frac{(1 - e^{-\lambda_i T_{1,r}})q}{1 - (1 - e^{-\lambda_i T_{1,r}})(1 - q)} \quad (5)$$

本文是通过目标性能需求来估算缓存容量,预先设置了第一层节点的目标命中率 $p_{1,r}$ ,第一层节点 r 的特征时间 $T_{1,r}$ 可通过不动点方程式(5)求得。此时,节点 r 的缓存容量 $C_{1,r}$ 可以通过下列公式(6)计算得出:

$$C_{1,r} = \sum_{i=1}^N \frac{(1 - e^{-\lambda_i T_{1,r}})q}{1 - (1 - e^{-\lambda_i T_{1,r}})(1 - q)} \quad (6)$$

### 4.2 第二层节点缓存容量分析

接下来分析第二层节点缓存的命中情况,首先计算第一层节点的等价特征时间 $T'_1$ 。根据文献[9]中提出

了的丢失率的概念: $m_{i,l,r}$ 为内容 i 在第 l 层第 r 个节点上的丢失率, $m_{i,l,r} = 1 - p_{hit}(i, l, r)$ , $m_{i,l}$ 为内容 i 在第 l 层的丢失率,计算为:

$$m_{i,l} = \sum_{r=1}^R m_{i,l,r} \frac{\sum_{i=1}^N \hat{\lambda}_i m_{i,l,r}}{\sum_{r=1}^R \sum_{i=1}^N \hat{\lambda}_i m_{i,l,r}} \quad (7)$$

此时可以得到第一层的等价命中率:

$$\tilde{p}_1 = 1 - \sum_{i=1}^N \hat{\lambda}_i \cdot m_{i,1} \quad (8)$$

在第一层,节点的命中率等于存在概率,将得到的等价命中率代入式(5)解不动点方程即可得到第一层节点的等价特征时间 $T'_1$ 。

根据系统命中率和各层命中率的关系,可以计算出第二层节点的目标命中率:

$$\tilde{p}_2 = 1 - \frac{1 - \tilde{p}_{sys}}{1 - \tilde{p}_1} \quad (9)$$

内容 i 在第二层命中,则意味着它在第一层没有被缓存。为了方便计算,假设请求到达第二层还是泊松过程,令 $\bar{\lambda}_i$ 为请求 i 到达第二层缓存的平均到达率,这里近似 $\bar{\lambda}_i = \lambda_i \cdot m_{i,1}$ 。对于 q-LRU 而言,两种情况:

i)当第二层的特征时间 $T_2 > T'_1$ 时,内容 i 的请求在第二层命中等价于 i 的前一次请求在第二层命中或者是未命中并以 q 的概率插入缓存中,并且该请求到达第二层缓存的时间位于 $[t - T_2, t - T'_1]$ 间隔内,或位于 $[t - T'_1, t]$ 间隔但没有在第一层缓存对应的内容。由此,可以得到内容 i 在第二层节点的命中率近似为:

$$p_{hit}(i, 2) \approx [p_{hit}(i, 2) + q(1 - p_{hit}(i, 2))] \cdot (1 - e^{-\bar{\lambda}_i(2)(T_2 - T'_1)} \cdot e^{-\bar{\lambda}_i(2)(1-q)T'_1}) \quad (10)$$

其中 $[p_{hit}(i, 2) + q(1 - p_{hit}(i, 2))]$ 表示 i 的前一次请求已存在第二层缓存中,或者不存在并以 q 的概率插入第二层缓存中; $(1 - e^{-\bar{\lambda}_i(2)(T_2 - T'_1)} e^{-\bar{\lambda}_i(2)(1-q)T'_1})$ 表示该请求到达第二层缓存的时间位于 $[t - T_2, t - T'_1]$ 间隔内,或位于 $[t - T'_1, t]$ 间隔但没有在第一层缓存对应的内容。

ii)当第二层的特征时间 $T_2 < T'_1$ 时,内容 i 要在第二层命中,则 i 的前一次请求在第一层没有缓存,此时,内容 i 在第二层的命中率可近似为:

$$p_{hit}(i, 2) \approx [p_{hit}(i, 2) + q(1 - p_{hit}(i, 2))] \cdot (1 - e^{-\bar{\lambda}_i(2)(1-q)T_2}) \quad (11)$$

内容 i 在第二层的丢失率 $m_{i,2} = 1 - p_{hit}(i, 2)$ 。并且,系统命中率和丢失率的关系为:

$$\tilde{p}_{sys} = 1 - \sum_{i=1}^N \hat{\lambda}_i \prod_{l=1}^L m_{i,l} \quad (12)$$

将式(12)代入(9),可以得到第二层的整体命中率为:

$$\tilde{p}_2 = 1 - \frac{\sum_{i=1}^N \hat{\lambda}_i \cdot m_{i,1} \cdot m_{i,2}}{\sum_{i=1}^N \hat{\lambda}_i \cdot m_{i,1}} \quad (13)$$

类似于第一层缓存的估算,将式(9)得到的第二层目标命中率  $\tilde{p}_2$  带入(13)解不动点方程,计算出第二层缓存的特征时间  $T_2$ 。

不同于第一层缓存,在第二层缓存中,命中概率是小于存在概率的。内容  $i$  在第二层的存在概率为:

$$p_{in}(i, 2) = (1 - e^{-\bar{\lambda}_i T_2}) [p_{in}(i, 2) + q(1 - p_{in}(i, 2))] \quad (14)$$

最后,第二层节点的缓存容量为:

$$C_2 = \sum_{i=1}^N p_{in}(i, 2) \quad (15)$$

至此,缓存容量估算完毕,算法伪代码如下:

算法:基于“Che近似”的CCN缓存资源估算算法

输入:  $p_{1,r}, \tilde{p}_{sys}, \lambda, R, \alpha, N$

输出: 缓存容量集合  $\{C_{l,r}\}$

```

1  for node r = 1 to R in level 1 do
2      for content i = 1 to N do
3          caclulate  $T_{1,r}$  by (4), (5)
4      end for
5      caclulate  $C_{1,r}$  by (6)
6      for content i = 1 to N do
7          caclulate  $m_{i,1,r}$  by (4)
8      end for
9      caclulate  $m_{i,1}$  by (7)
10     end for
11     caclulate  $\tilde{p}_1 \rightarrow T'_1$  by (8), (5)
12     caclulate  $\tilde{p}_2$  by (9)
13     for content i = 1 to N do
14         caclulate  $T_2$  by (10) - (13)
15     end for
16     caclulate  $C_2$  by (15)

```

### 4.3 计算结果分析

本文的估算方法在第一层缓存上假设内容流行度服从Zipf分布,到达率过程为泊松过程,使用“Che近似”理论,使用目标命中率来估算缓存容量,估算结果是较为准确的。

对于第二层缓存的估算,假设请求到达第二层的过程还是简单的泊松过程,但实际上,请求到达第二层缓存时是一个类似于开关调制的过程(ON-OFF),其中ON状态表示内容  $i$  没有缓存在第一层缓存中,因此被转发到第二层缓存,而OFF状态表示内容  $i$  存在于第一层缓存,对  $i$  的请求不能到达第二层缓存。为了估算出第二层的缓存容量,本文近似将其看作泊松过程,进而得到第二层缓存的特征时间  $T_2$ ,并且估算出第二层的缓存容量  $C_2$ 。但是这样计算出来的结果可能会有一定的误差。

## 5 仿真实验

本文使用ndnSIM进行仿真实验,来评估估算方法的准确性,并分析了不同参数下的估算方法性能。实验使用图1所示的网络拓扑,假设第一层有4个缓存节点,节点的缓存替换策略为q-LRU,内容的流行度服从Zipf分布,节点的内容请求速率为每秒100次请求,为了方便计算,假设每个内容块的大小均为1。若无特殊说明,系统命中率设为0.5,网络中内容的数量  $N$  为10000,第一层为同构缓存,命中率为0.25,Zipf参数  $\alpha$  为0.8,缓存概率  $q = 0.5$ 。

### 5.1 缓存概率 $q$ 的影响

图2和图3展示缓存概率  $q$  对估算方法和缓存容量的影响。如图2所示,随着缓存概率  $q$  的变化,仿真得到的命中率保持着稳定状态,由此可以得知缓存概率  $q$  基本不影响估算方法的准确性,即改变  $q$  的大小,估算方法都能得到较为准确的结果。

图3显示了相比于传统的LRU策略 ( $q = 1$ ),当缓存概率  $q = 0.1$  时,第一层每个节点的缓存容量减少了约43.2%,第二层节点的缓存减少了约34.7%,系统总缓存减少了约38.2%。结合图2,可以得出:随着缓存概率  $q$  的减小,在不影响网络性能的同时,可以有效减小缓存容量,提高资源利用率。说明  $q < 1$  时, q-LRU 的缓存性能比 LRU 更好。

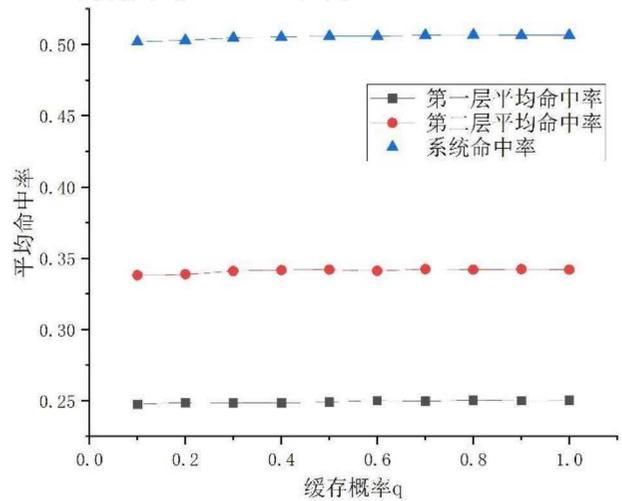


图 2:缓存概率  $q$  对缓存命中率的影响

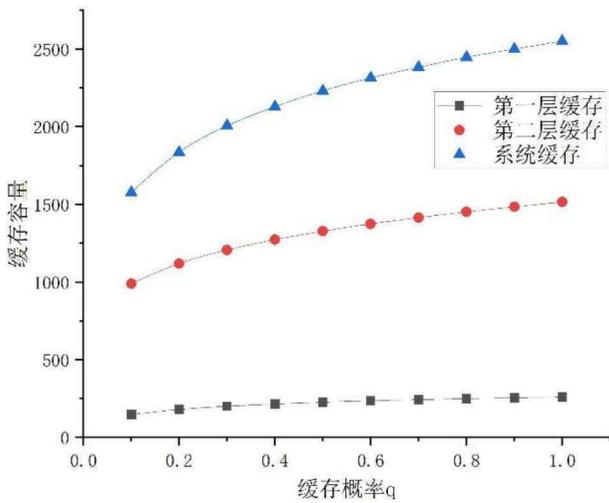


图3:缓存概率q对缓存容量的影响

### 5.2 Zipf 参数 $\alpha$ 的影响

图4和图5展示Zipf 参数 $\alpha$ 从0.6到1.2对估算方法和缓存容量的影响。图4中展示了Zipf参数 $\alpha$ 变化时，各层平均命中率以及系统命中率的变化，从图中可以看出，当 $\alpha = 1.2$ 时，结果有了明显的波动，这是因为此时流行内容占大多数，只需很小的缓存便可以满足需求，此时分配给第一层的缓存极小仅为5，而计算的结果约为5.13，因此出现了波动。但当 $\alpha$ 的取值稍微小时，结果比较稳定，可以得出 Zipf 参数 $\alpha$ 对估算方法的准确性影响较小。

此外，图5说明了缓存容量受流行度的影响较大，随着Zipf参数 $\alpha$ 的增加，缓存容量呈指数下降，当 $\alpha$ 从0.6加到1.2时，在不改变目标命中率的同时所需的缓存资源降低了约140倍。

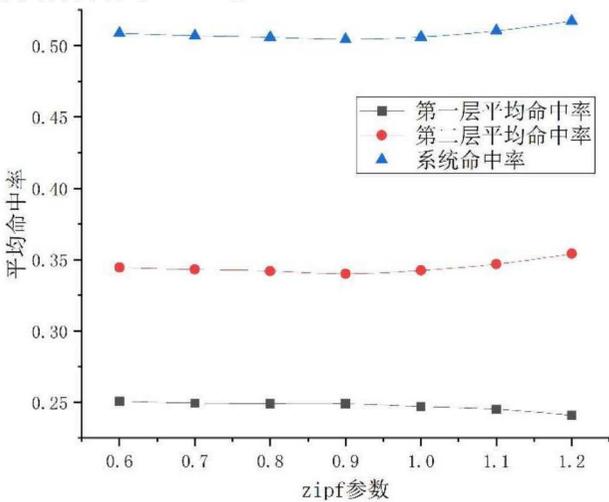


图4:Zipf 参数 $\alpha$ 对缓存命中率的影响

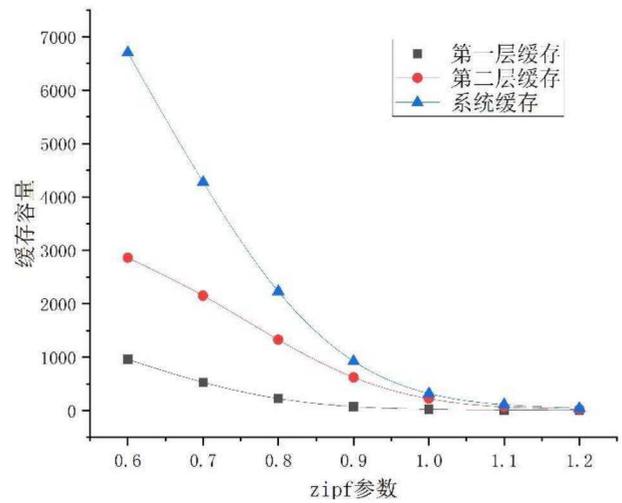


图5:Zipf 参数 $\alpha$ 对缓存容量的影响

### 5.3 内容数量N的影响

图6和图7显示了改变网络中内容数量对估算方法和缓存容量的影响，图6是固定系统命中率的情况下对各个内容数量做的仿真实验，从图中可以看出，仿真结果的命中率随内容数量的增加，波动很小，因此可以得出估算方法的准确性基本不受网络内容数量的影响。

图7展示了缓存容量随内容数量变化的关系，可以看到随着缓存容量的增加，要想实现同样的性能，缓存容量的增加是线性的。

图8展示了固定缓存容量的情况下，网络中的命中率随网络内容数量增加而改变，从结果可以得出，内容数量的增加和命中率的降低是不成比例的，图中 $N=10000$ 到 $N=100000$ ，网络中的命中率降低趋势越来越小，两者是非线性变化的。结合图7和图8，网络中内容数量对缓存性能有影响，但是并不是主要影响因素。

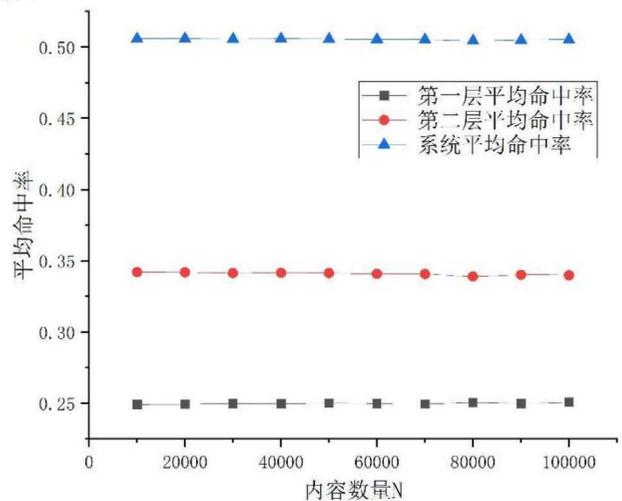


图6:内容数量 N 对缓存命中率的影响

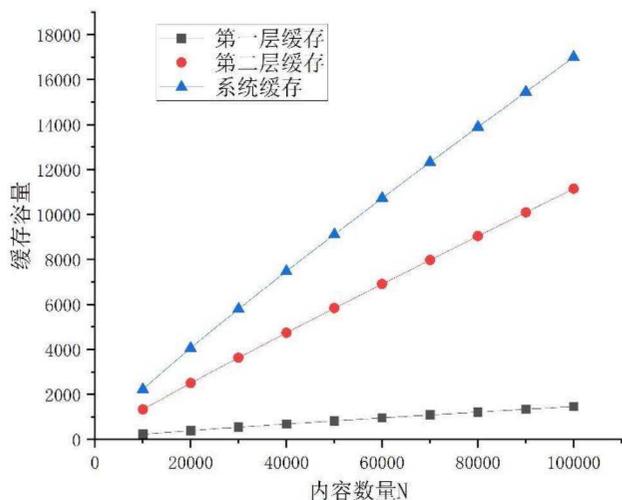


图7: 内容数量 N 对缓存容量的影响

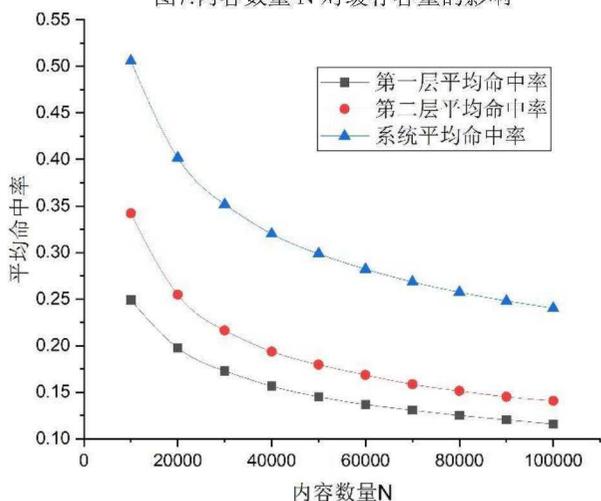


图8: 内容数量 N 对平均命中率的影响

#### 5.4 第一层节点命中率的影响

图9和图10是改变第一层平均命中率对估算方法和缓存容量的影响。如图9所示，第一层节点的命中率的实验设置和仿真结果几乎完全拟合，证明估算方法在第一层是准确的。而对于第二层来说，实验设置和仿真结果具有明显的误差，这是因为估算方法在第二层是假设请求到达过程依旧是泊松过程，而实际上，请求到达第二层缓存时是一个类似于开关调制的过程(ON-OFF)，这是第二层误差的主要原因。而对于系统命中率而言，第二层的误差是引起系统误差的主要原因。

图10是第一层平均命中率对缓存容量的影响，从图中可以看出， $q = 0.5$ 和 $q = 1$ 时缓存容量的变化趋势相同，随着第一层平均命中率的增加，第一层节点的缓存容量也在增加，而第二层的缓存容量是先增后减，主要是因为当第一层缓存容量较小时，只能满足少量请求，多数的请求将被转发到第二层；而随着第一层缓存容量的增加，多数的请求可以在第一层命中，因此第二层的缓存容量开始减少。图10中总的缓存容量是不断增加的，如果只考虑缓存资源，将多数缓存分配给第二层节点可以有效节约缓存资源，但是小的

第一层缓存将会转发大量请求到第二层，增加系统中传输的流量，结果会使网络成为类似传统的IP网络。

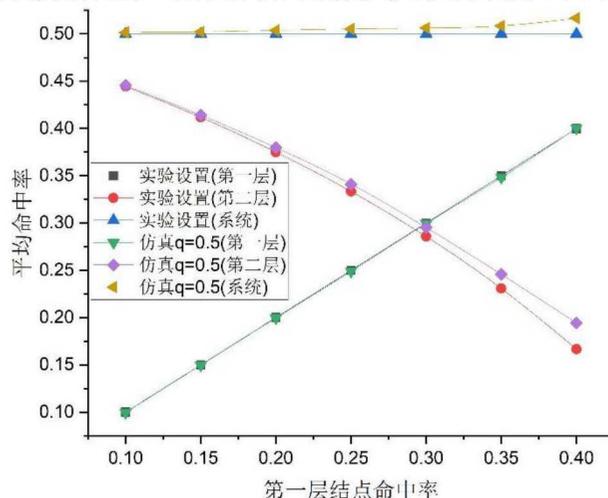


图9: 第一层平均命中率对缓存命中率的影响

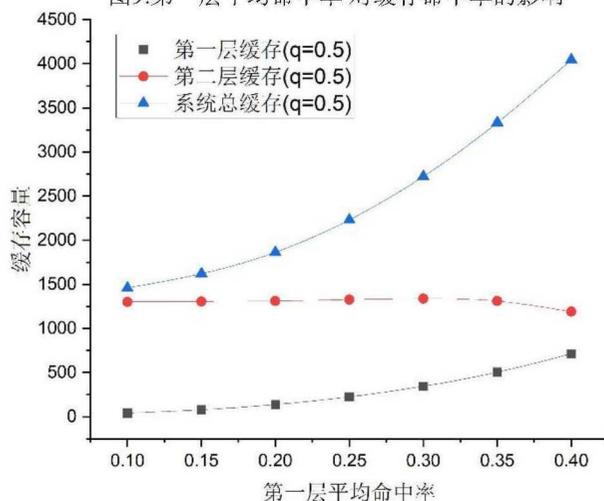


图10: 第一层平均命中率对缓存容量的影响

从这个实验中可以分析得到，当第一层节点是异构缓存时，本文的估算方法同样适用，因为这个估算方法的主要误差并不在第一层节点，相反，第一层节点的拟合程度是非常高的。因此，对于异构缓存而言，只需要分别计算第一层每个节点的缓存容量再通过计算第一层等价特征时间，就可以估算出第二层的容量大小。

## 6 结论

本文考虑到当前 CCN 网络还无法大规模部署应用这一现状，因此以小规模接入网为切入点，探究在网络接入层引入小规模 CCN 网络架构，所需要的缓存资源，以及通过引入缓存，量化其带来的网络服务能力。为此，本文以网络拓扑的系统平均缓存命中率为目标，根据“Che 近似”理论及其扩展研究，分析了层次缓存网络中的内容特性，并以此提出了一个两层树状拓扑节点缓存资源估算方法，根据节点缓存的目标命中率，估算网络中各个节点所需的缓存容量。并通过实验仿真，验证了估算方法的准确性，可以为网络运营商规划和部署CCN网络提供参考。

## 参考文献

- [1] Cisco Visual Networking Index: Forecast and Methodology 2017 - 2022, San Jose, CA, USA: Cisco, Nov, 2018.
- [2] Jacobson V, Smetters D K, Thornton J D, et al. Networking named content[C]. ACM CoNEXT' 09, 2009:1-12.
- [3] Jacobson V, Smetters D, Thornton J, et al. Networking named content. In CoNext 2009, 2009.
- [4] Cui X, Liu J, Huang T, et al, A novel metric for cache size allocation scheme in Content Centric Networking[C] National Doctoral Academic Forum on Information and Communications Technology 2013: 1-6.
- [5] Tang Y. Minimizing Energy for Caching Resource Allocation in Information-Centric Networking with Mobile Edge Computing. 2019 IEEE Intl Conf on Congress (DASC/PiCom/CBDCCom/CyberSciTech), Fukuoka, Japan, 2019. 301-304.
- [6] Yuan D, Xu Y, Ran J, et al. An optimal fair resource allocation strategy for a lightweight content-centric networking architecture. 2017 3rd IEEE International Conference on Computer and Communications (ICCC), Chengdu, China, 2017. 573-577.
- [7] H. Che, Y. Tung and Z. Wang, "Hierarchical web caching systems: modeling design and experimental results", IEEE JSAC, vol. 20, no. 7, pp. 1305-1314, Sep 2002.
- [8] M. Garetto, E. Leonardi and V. Martina, "A unified approach to the performance analysis of caching systems", ACM Trans. Model. Perform. Eval. Comput. Syst., vol. 1, no. 3, pp. 12:1-12:28, May 2016.
- [9] Rizk A, Zink M, Sitaraman R, Model-based design and analysis of cache hierarchies[C]. IFIP Networking Conference (IFIP Networking) and Workshops, Stockholm, 2017:1-9.
- [10] Tyson G, Kaune S, Miles S, et al. A Trace-Driven Analysis of Caching in Content-Centric Networks[C] 21st International Conference on Computer Communications and Networks (ICCCN), 2012: 1-7.
- [11] Chiochetti R, Rossi D, Rossini G, et al. Exploit the known or explore the unknown?: hamletlike doubts in icn[C] Proceedings of the second workshop on Information-centric networking, 2012: 7-12.
- [12] Fricker C, Robert P, Roberts J, et al. Impact of traffic mix on caching performance in a contentcentric network[C] Proceedings IEEE INFOCOM Workshops, 2012: 310-315.
- [13] Rossi D, Rossini G. On sizing CCN content stores by exploiting topological information[C], Proceedings IEEE INFOCOM Workshops, 2012: 280-285.
- [14] Wang Y, Li Z, Tyson G, et al, Design and Evaluation of the Optimal Cache Allocation for Content-Centric Networking[J], IEEE Transactions on Computers, 2016, 65(1): 95-107.
- [15] Wang Y, Li Z, Tyson G, et al, Optimal cache allocation for Content-Centric Networking[C], 21st IEEE International Conference on Network Protocols (ICNP), 2013: 1-10.