

基于聚类算法内容流行度预测的空天地一体化网络缓存方法

仇超¹, 王晨阳^{2,3}

(1. 天津大学, 天津 300072;

2. 深圳大学计算机与软件学院, 广东 深圳 518060;

3. 人工智能与数字经济广东省实验室(深圳), 广东 深圳 518132)

摘要: 针对空天地一体化网络中恶劣的自然条件可能造成连通异常和通信阻塞的情况, 对一种新的空天地一体化网络缓存方法进行研究, 即基于聚类的内容缓存方法。基于空天地一体化网络架构, 利用提取的内容特征来预测流行度, 并用于缓存替换决策。内容的流行度预测是由动态聚类算法实现的, 这种方法可以有效地减轻地面核心网的通信压力, 从而进一步提升用户体验。使用开放数据集对算法进行评估, 实验仿真结果证明了该方法的可行性和有效性。

关键词: 空天地一体化网络; 内容缓存; 流行度预测; 聚类算法

中图分类号: TP393

文献标识码: A

doi: 10.11959/j.issn.2096-8930.2024005

A Space-Air-Ground Integrated Network Caching Method Based on Clustering Algorithm Content Popularity Prediction

QIU Chao¹, WANG Chenyang^{2,3}

1. Tianjin University, Tianjin 300072, China

2. College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China

3. Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ), Shenzhen 518132, China

Abstract: This paper proposed a new space-air-ground integrated network caching method, that was, cluster based content caching method. Based on the space-air-ground integrated network architecture, this method used the extracted content features to predict the popularity and used it for cache replacement decision. Content popularity prediction was realized based on dynamic clustering algorithm. This method could effectively reduce the pressure on the ground backbone network and further improve the quality of user experience. It evaluated the performance of our method using the open Douban movie review dataset. This dataset could be used to simulate user requests. Experimental simulation results showed the feasibility and effectiveness of the method.

Keywords: space-air-ground integrated network, content caching, popularity prediction, cluster algorithm

0 引言

新一代通信技术的快速发展, 与互联网、大数据、人工智能等流行信息技术相结合, 催生了移动支付、网络游戏、远程医疗、无人驾驶等众多新型移动应用。此外, 通信技术还在工业、交通、医疗、教育等领域得到应用, 有效促进了生活中方方面面的信息化和数字化转型^[1]。随着地面移动通信的快速发展, 具有智能感知能力的物联网也

得到了前所未有的发展。爱立信移动报告显示^[2], 到 2027 年, 大约 2.3 亿台终端设备每月将产生 288 艾字节 (EB) 的数据, 这对网络架构和先进通信技术的革命性发展提出了要求。同时, 随着网络上多媒体业务数量和质量的不断提高, 用户设备的流量和计算量急剧增加, 给已经拥堵的骨干网增加了巨大的工作量。

在架构方面, 地面通信网络虽然可以配置强大的软硬件资源来提供高速通信, 仍难免因恶劣的自然条件、地质

收稿日期: 2023-12-01; 修回日期: 2024-03-01

基金项目: 国家自然科学基金青年项目资助 (No.62002260)

Foundation Item: The National Natural Science Foundation of China (Youth) (No.62002260)

灾害等因素造成连通性异常和通信阻塞,最终导致通信覆盖盲区。因此,具有广域范围和宽带通信能力的空间与卫星综合通信成为提供不受地域限制、无缝覆盖和连接的前瞻性技术。包括 SpaceX、中国卫星网络集团有限公司在内的国内外公司都启动了发射巨型低地球轨道(Low Earth Orbit, LEO)卫星的计划,未来将建立的服务对互联网接入至关重要^[3]。然而,带宽和能源的限制导致与地面基站相比,该方式可部署的计算和通信资源有限,周期性的轨道飞行往往需要星间通信来实现覆盖中继。为更有效地满足地面用户的应用需求,构建多域互联协作的多层网络架构,对提供异构空间服务具有重要意义。

空天地一体化网络(Space-Air-Ground Integrated Network, SAGIN)是集卫星系统、航空系统、地面通信系统为一体的跨域异构系统,为各域用户终端提供灵活的节点接入、高速数据传输和无缝覆盖。SAGIN的互联性能超出了传统单一网络形式,引起了全世界研究人员的广泛关注^[4]。例如,欧洲航天局近期宣布,第五代通信链路下的欧洲与日本间基于SAGIN架构的洲际连通性首次试验成功,其还计划开发连接多个节点的技术,例如卫星、无人驾驶飞行器和高空平台,以获得无处不在的连接^[5]。

在技术方面,边缘缓存通过网络热点内容在边缘服务器的重用可以有效减少网络的负载时间、减轻服务器的压力。相较于内容分发网络,边缘缓存的应用可以进一步提高超高清视频等服务的用户体验,已成为5G的重要技术。借鉴地面通信范畴中终端-边缘-云中心3层架构的部署基础,边缘缓存存在空天地一体化网络中实现的难度是可控的^[6]。

改良缓存算法是改善用户体验、提升缓存性能和效率的有效途径^[7],缓存性能提升的挑战主要来自缓存容量。单个边缘缓存节点或个人设备的存储量相对于核心网络非常有限,这是导致低缓存命中率的重要因素。为了提高内容缓存的效率,大量研究工作致力于从路径策略、服务器布局、内容复制策略和其他方面进行优化^[8]。

缓存系统的性能高度依赖于高效的缓存策略。然而,很少有人关注缓存策略的改进问题。传统的缓存置换策略,如最近最少使用(Least Recently Used, LRU)策略和先进先出(First In First Out, FIFO)策略,虽然实现简单,但它们基于静态规则,可能会忽略用户需求的变化,从而导致缓存资源的浪费。随着人工智能技术的突破,更多的研究内容逐渐偏向于动态自适应缓存方法的设计。这样的调度算法可以最大限度地利用缓存节点的存储容量,从而进一步减轻主干网络的压力。本文的主要贡献如下。

(1)提出了基于聚类的内容缓存算法,该算法提取用户请求的内容特征并预测其流行度。根据流行度预测的结果,该算法可以做出更好的缓存决策,以减少对地面核心网络的请求数量。

(2)使用聚类算法来实现内容流行度的动态分析和预测,可以使缓存方法根据用户需求的变化而进行调整,从而基于当前情况和未来预测做出最佳缓存决策。

(3)同一些最先进的基准算法比较,实验结果表明了所提算法的优越性。实验部分使用了来自某网站的电影评论数据,该数据可用于模拟用户请求。

1 相关工作

随着5G网络的发展,众多研究以优化移动网络架构和改善用户体验为目的,集中于网络路由、分布式服务、服务器布局等方面。例如,参考文献[9]提出了一种遗传算法用以解决最短路径路由问题。参考文献[10]提出了轻量级协作缓存管理算法,旨在最大化缓存服务的流量并最小化带宽成本。参考文献[11]描述了如何利用从社交联系中提取的地理信息来改进内容交付网络中多媒体文件的缓存。

现有的优化工作更侧重于内容缓存的架构级别,忽略了缓存替换策略也存在优化空间。参考文献[12-13]总结了各种常用的缓存方案。最常用的缓存方案,如先进先出策略和最不经常使用(Least Frequently Used, LFU)策略虽然稳定但不能利用时间和内容特性进行优化,而更加智能的缓存策略需要综合考虑历史和未来条件。参考文献[14]提出了通过机器学习提高缓存性能。参考文献[15]给出了智能代理缓存方法,支持向量机和决策树算法(C4.5)应用于日志文件中。

内容流行度模型是一个确定每条内容的流行度的函数,即每条内容被请求的频率。一些研究集中于通过机器学习预测内容的流行度。例如,参考文献[16]提出了一种在线算法,该算法可以有效地捕获和维护通过缓存代理请求的Web对象准确的流行概况。参考文献[17]提出了一种在线算法,该算法可以学习内容的未来流行度与其最近的访问模式之间的关系。

2 缓存模型

在空天地一体化网络中,一般存在3层架构,共同实现多空间、多网域的异构融合和跨域互联。它们可以独立工作,也可以互通。其中,最上层为由低地球轨道卫星、中地球轨道(Medium Earth Orbit, MEO)卫星和地球静止轨道(Geostationary Earth Orbit, GEO)卫星所组成的空间

段。随着轻型卫星技术、相控阵多波束天线、星上实时处理、星间链路等技术的发展, LEO 卫星变得更加经济化、小型化, 并可以部署丰富的资源。在 SAGIN 系统中, 本文使用具有计算和存储能力的 LEO 卫星作为边缘服务器节点, 来完成边缘缓存处理任务, 即通过部署缓存算法来实现内容调度并对终端提供服务。

在空中段, 数据采集、传输和任务处理主要由无人机、高空平台等载体完成。它们的传输时延比卫星短, 覆盖范围比地面基站广, 具有部署灵活、成本低、响应快、续航能力强等特点。空中段网络可以实现低时延和广域覆盖, 但存储容量有限, 链路不稳定。

地面段包括物联网设备、用户终端、通信站、地面基站和其他支持地面服务的基础设施。地面基站计算能力强, 能源供给充足, 配置在地面基站的边缘服务器可以提供低时延、高效率的就近服务, 但易受地形限制和部署约束。

边缘缓存节点的架构如图 1 所示。除了所有缓存节点都包含的常见模块, 例如请求接收器、内容发送器、请求处理器、内容管理器、内容获取器、本地缓存, 本文在原始架构中引入了新的模块, 即特征提取器、聚类管理器、流行度查询器和决策器, 以控制缓存过程。

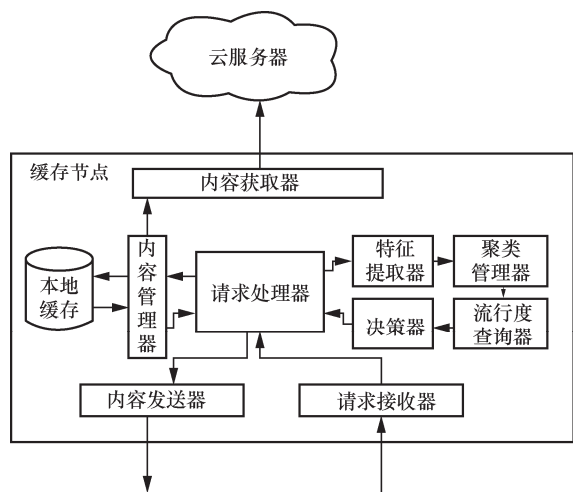


图 1 边缘缓存节点架构

内容发送器将请求的内容发送回用户终端, 请求接收器则是接收用户的请求并将其发送给请求处理器。本地缓存负责将内容存储在本地缓存列表中。此外, 内容管理器管理本地缓存列表中的内容和来自云服务器的内容。请求处理器作为一个中间模块, 协调不同模块和整个缓存工作流之间的工作。特征提取器负责从原始请求中提取特征, 聚类管理器负责聚类的更新和动态拆分, 流行度查询器基于特征和聚类的结果对所请求的内容进行预测。决策器维护一

个优先队列的数据结构, 以动态记录缓存列表中各项内容的预期流行度, 并做出是否缓存新内容的决策。内容获取器负责与云服务器通信以获取不在本地缓存列表中的内容。

请求服务的流程如下。

首先, 请求接收器接收新的请求, 并对该请求进行预处理, 以提取关于该请求的信息, 如内容 ID、用户 ID、请求时间等。接下来, 传输此信息到请求处理器。请求处理器调度内容管理器从本地缓存列表中获取内容。这里有两种情况。

第一种, 内容在缓存列表中。请求处理器将请求内容返回, 记录访问日志, 请求可以立即得到满足。

第二种, 内容不在缓存列表中。内容管理器将请求发送到内容获取器模块, 后者与云服务器通信以获取内容并将其返回用户。请求处理器首先将请求传输到特征提取器, 以开始缓存决策过程。请求处理器告知内容管理器执行缓存决策。如果选择缓存新内容, 当缓存列表未满时, 内容管理器可以直接执行。当缓存列表已满时, 请求处理器将缓存决定发送给内容管理器, 以基于流行度排序用新内容替换旧内容; 否则, 缓存列表中没有任何更改。

其中, 缓存决策过程遵循以下步骤。

特征提取器接收内容并提取预测所需的特征, 然后将特征向量以键值对的形式保存到特征数据库中, 以便后续复用。

流行度查询器接收特征提取器的输出, 并将聚类结果存储在聚类数据库中。然后, 预测新内容的预期流行度, 并将其发送给决策者。

决策器维护优先队列, 以记录缓存列表中的预期流行度, 并基于新内容的预期流行度做出是否缓存新内容的缓存决策。

假设有 $F=\{1,2,\dots,F\}$ 个大小相同的可访问内容, 始终保存于云服务器 (注: 在实际的缓存应用场景中也可以满足相同的大小假设。在这些场景中, 可以将原始内容分割成大小相同的小块, 每个小块都可被视为一个可访问的内容^[18-20])。为了提高移动用户的服务质量 (Quality of Service, QoS) 或体验质量 (Quality of Experience, QoE), 内容提供商必须设置缓存系统以将请求卸载到边缘缓存节点。通过这种方式, 移动用户可以直接从边缘缓存节点获取内容, 不仅避免了与云服务器的长通信时延, 还释放了主干网络的压力。

本文仅考虑单一缓存节点并假设系统中各个缓存节点是相互独立的。定义 $C < F$ 为缓存节点的最大缓存内容数, $R = \{r_1, r_2, \dots, r_m, \dots, r_M\}$ 是缓存节点接收的请求。每个请求

可以被描述为一个三元组 $r_m = (f_m, t_m, a_m)$, $\forall 1 \leq m \leq M$, 其中 $f_m \in F$ 是内容 ID, t_m 是请求时间, a_m 是 d 维的向量, 用来表示请求特征, 例如请求内容类型和用户情况。本文将特征向量归一化且 $a_m \in [0, 1]^d \triangleq A$ 。

定义 $S_m = [S_{m(1)}, S_{m(2)}, \dots, S_{m(F)}]$ 为 t_m 时刻的缓存列表状态, 其中 $S_{m(k)} \in \{0, 1\}$ 是第 k 项内容的缓存状态。 $S_{m(k)} = 1$ 表示第 k 项内容当前缓存在边缘节点, 对其的请求可以直接得到响应, $S_{m(k)} = 0$ 则相反, 此时内容 f_m 不存在于边缘节点, 需要向云服务器请求并决定是否进行缓存, 尤其是当边缘节点缓存已满时, 边缘节点必须进行缓存替换。定义 f_m^{old} 为被替换内容, 因此, t_m 到 $t_m + 1$ 的缓存状态变化可以表示为

$$S_{m+1}(k) = \begin{cases} 0, & k = f_m^{\text{old}} \\ 1, & k = f_m \\ S_m(k), & \text{其他} \end{cases} \quad (1)$$

为了提高移动用户的 QoS 和 QoE, 应设计一个有效的缓存策略, 以便在 R 中的请求到达时做出适当的缓存决策。缓存策略可以表示为从当前缓存状态、请求的内容和内容的特征向量到下一个缓存状态的映射 $\pi: (S_m, F, A) \rightarrow S_{m+1}$ 。当新请求到达时, 缓存状态基于策略 π 变化

$$S_{m+1} = \pi(S_m, f_m, a_m) \quad (2)$$

通常, 缓存命中率 $H(\pi, M)$ 是评价缓存系统效率的主要评价指标^[21], 定义为请求直接响应数占有所有 M 个请求的百分比。本文使用 $H(\pi) = E[H(\pi, m)]$ 表示策略 π 的预期平均命中率, 其计算过程可以表示为

$$H(\pi) = \lim_{M \rightarrow \infty} \frac{1}{M} \sum_{m=1}^M H(\pi, m) = \lim_{M \rightarrow \infty} \frac{1}{M} \sum_{m=1}^M \frac{\sum_{k=1}^m S_k(f_k)}{m} \quad (3)$$

这样, $H(\pi)$ 就代表了缓存策略 π 的预期性能。对于缓存系统, 期望找到一个最优策略 $\pi^* \in \Pi$ 使期望命中率 $H(\pi)$ 最大化, 可以表示为

$$\pi^* = \operatorname{argmax}_{\pi \in \Pi} H(\pi) \quad (4)$$

3 流行度预测算法

当本地缓存节点接收到新请求时, 首先检查内容是否在其本地缓存列表中。如果是, 将从本地缓存列表中获取内容, 并立即为用户服务; 否则, 会从云服务器下载内容, 然后将其发送给用户。之后, 本地缓存节点决定是否将其缓存在本地缓存列表中^[22]。

首先, 提取所请求内容 f_m 的本身特征。接下来, 根据

内容的历史访问特征求得请求特征 a_m , 预测流行度 $\tilde{P}_m$ 。 $\tilde{P}_m$ 将与本地缓存列表中内容的流行度进行比较。为了加快搜索过程, 保留一个优先队列 Q , 以便只能将 $\tilde{P}_m$ 与 Q 中的顶部元素 $(\tilde{P}^{\text{least}}, f^{\text{least}})$ 进行比较。如果 $\tilde{P}_m > \tilde{P}^{\text{least}}$, 则边缘缓存节点将从其缓存列表中删除顶部元素 f^{least} ; 然后插入 $(\tilde{P}_m, f_m)$ 来更新优先队列, 否则, 一切都不会改变; 最后, 存储 f_m 的访问日志^[23]。

同时, 为了及时保持内容的预期流行度, 要定期更新优先队列。这样随着热门内容的变化, 内容的预期流行度将发生变化, 算法可以随时跟踪不同时间的热门内容。此外, 在某个请求 θ 之后, 可以在这段时间内获得 f_m 的流行度 P_m , 然后使用实际流行度来更新本文的预测模型。

为了进行适当的流行预测, 本文主要使用两种特性: 访问日志特性和与内容本身相关的特性。根据数据源, 可以选择不同的功能, 例如, 对于不同的电影包含的不同标签进行分类, 并且每个电影都有自己的访问历史。这些标签和访问历史可以被视为请求特征。本文使用了某网站电影评论数据, 该数据包含社区电影频道的用户评论。

对于新的请求 r_m , 本文首先搜索历史日志以提取 f_m 的访问特征 h_m , 这表示用户历史行为的特征, 例如, 该内容截至当前的请求数。 h_m 是表示内容特征的 x 维向量。然后, 提取表示内容特征的 y 维向量 v_m , 例如电影的评分。为了便于数据处理, 本文将 h_m 和 v_m 利用最大值和最小值进行归一化, 即 $h_m, v_m \in [0, 1]$ 。请求特征 a_m 是 h_m 和 v_m 的融合。

一旦得到了特征向量 a_m , 就可以建立 h_m 的预期流行度。首先, 得到当前的聚类结果, 其表示为 $C_m \triangleq (\{c_k\}, K)$, 其中 K 是聚类的数量, $\{c_k\}$ 是聚类中心集; 然后, 找到离当前请求向量 a_m 最近的聚类中心 c_m 。一旦选择了聚类中心, 就可以获得聚类的总流行度 $P(c_m)$ 、聚类中节点的数量 $N(c_m)$ 以及集群中心 c_k 与同一集群中其他节点之间的最大距离 D_k 。最后, 计算 c_k 和 a_m 之间的距离 d_m , 并将 f_m 的预期流行度返回为 $\frac{D_k \times P(c_k)}{d_m \times N(c_k)}$ 。

为了更新聚类, 本文针对每次确定的聚类中心 c_k , 通过式 (5) 更新总流行度和向量个数

$$\begin{aligned} P(c_m) &= P(c_m) + P_m \\ N(c_m) &= N(c_m) + 1 \end{aligned} \quad (5)$$

同时, 当一个新向量节点加入聚类时, 如果集群离散度 $D(c_k)$ 超过阈值 Φ , 则通过 K -means 算法将聚类 c_k 的节点集 c_k 分割为 k 个新聚类。

K -means ($k=2$) 的过程基于以下内容, 如图 2 所示,

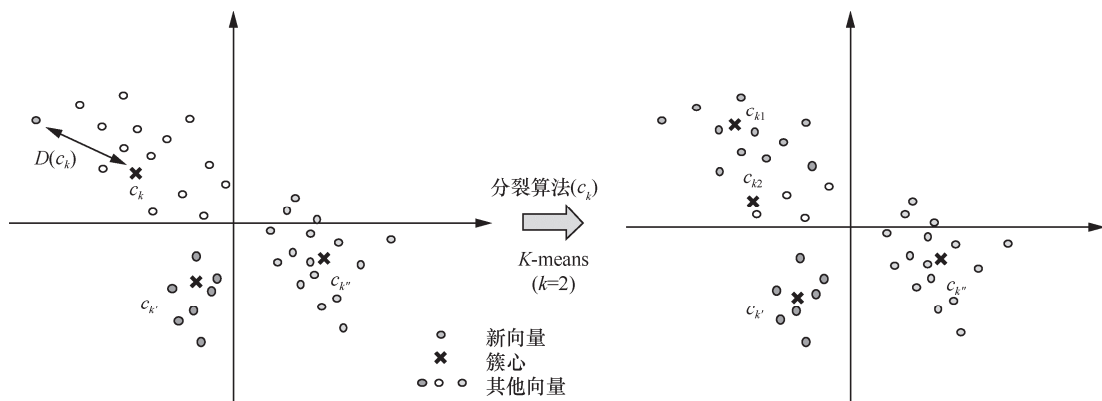


图 2 聚类分割示意

对于 k 取其他值也类似。首先, 随机选择两个节点 c_{diff1} 和 c_{diff2} 作为聚类 c_k 新的初始质心, 每个节点都属于节点和质心之间具有最小欧氏距离的簇。然后, 算法使用集群中的所有节点, 利用平均的方法重新计算质心。最后, 重复上述步骤, 直到质心不再改变。本文以两个新的聚类 c_k^1 c_k^2 更新聚类 c_k , 并在 K -means 算法完成时删除原有的 c_k 。

4 实验验证

本文使用从某网站电影频道爬取的数据作为算法评估的数据集。在本文的实验中, 假设有一个缓存节点向用户提供视频内容, 可以将某部电影的评论视为对某些内容的请求, 发布评论的时间是用户请求的时间。这种模拟过程可以在现实世界中得到充分的解释, 尤其是短视频应用程序: 随着内容的普及, 用户对相关内容的请求在一段时间内会增加, 然后逐渐减少并保持稳定。

为了获取数据, 本文采用爬虫程序来获取电影的 ID、评论的时间戳和用户的打分。为了保证数据集的正确性, 编写了随机爬虫来获取一个小数据集, 并与主数据集进行交叉验证。数据集一共包含 35 000 条不同的内容 ID, 200 000 条可见的评论, 评论时间的跨度为 2018 年 9 月 1 日至 2019 年 1 月 1 日。

本文设置实验中的请求特征向量有 4 个维度 ($d=4$), v_m 有 3 个维度 ($y=3$), 包括电影平均得分、电影的评论总数以及用户在此评论中对电影的评分。 h_m 有一个维度 ($x=1$), 即截至当前这部电影的评论总数。由于数据的尺度不同, 本文对数据进行了预处理。此外, 本文的算法中有一个参数, θ 设置为 5 000。

本文与以下先进的基准算法进行比较。

- (1) FIFO^[24]: 总是淘汰缓存中保存时间最长的内容。
- (2) LFU^[25]: 淘汰最不常用的内容。

(3) LRU^[26]: 选择最近未使用的内容进行淘汰。

(4) 基于动态时间老化的最少频率使用 (Least Frequency Used Dynamic Aging, LFUDA^[27]): 对象的命中次数和使用频率决定淘汰内容。

表 1 和表 2 显示了具有不同阈值 ϕ 下缓存命中率与 K -means 算法 k 值的关系。可以看出, 缓存命中率随着 k 的增加而降低。这是因为本算法使用不同聚类的特征进行流行度预测。因此, 较高的 k 意味着分裂出更多的新聚类。当 ϕ 不变时, P 和 N 在每个聚类中的分布会受到影响。更多的聚类意味着 $P(c_k)$ 和 $N(c_k)$ 不再突出聚类特征而是更平均。

表 1 不同 k 值下的缓存命中率 (ϕ 为 1.6)

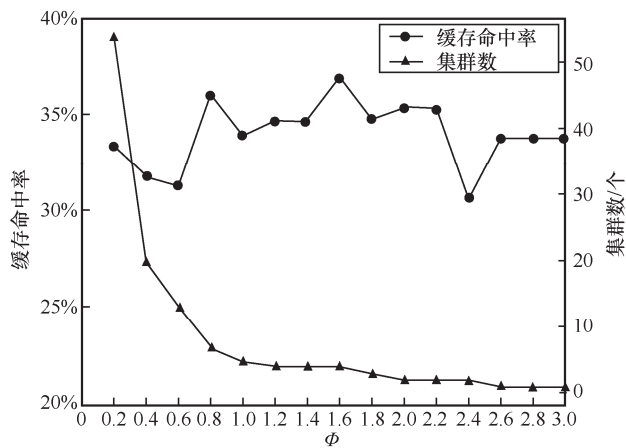
k	聚类数/个	命中率
2	4	36.9%
3	5	36.8%
4	7	36.1%
5	7	34.5%

表 2 不同 k 值下的缓存命中率 (ϕ 为 0.8)

k	聚类数/个	命中率
2	7	36.0%
3	13	35.7%
4	17	35.6%
5	21	33.8%

缓存命中率与集群数随 ϕ 的变化趋势如图 3 所示。有几点值得注意。① 对于这里的距离, 本文使用欧几里得距离的平方作为度量。由于本文对 4 个维度的特征进行了标准化, 因此该值的可选范围为 0~4。如果 ϕ 更大, 则执行分割算法的次数越多, 最终聚类的数量就越多。② 缓存容量保持不变, 缓存命中率随着 ϕ 值的增加而波动。对于数据集来说, 当 ϕ 值为 1.6 时, 缓存命中率可以达到最高的

36.9%；当该值为 2.4 时，缓存命中率最低，只有 30.54%，但仍比 LRU 的表现高 0.5%。该算法将所有请求的特征向量划分为两个聚类。



图注：缓存容量为 2 500 MB

图 3 不同 ϕ 值下缓存命中率和集群数

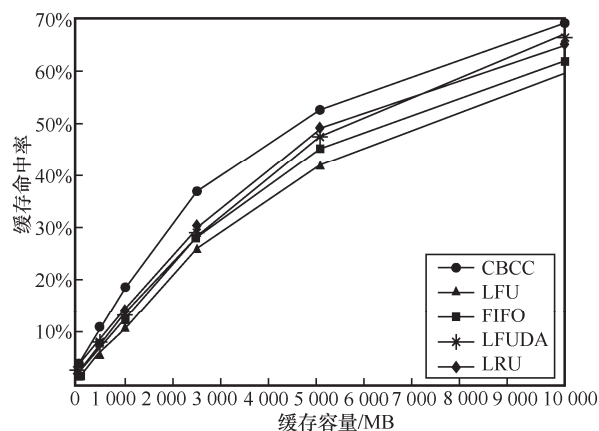
本算法和基准算法在各种缓存容量下的平均缓存命中率如图 4 所示。可以看出，本算法在所有模拟中显著优于其他算法，这是因为其他基准算法在做出缓存决策时没有考虑内容的未来流行度。其他基准算法只考虑内容当前的流行度，这可能与未来的流行度不同，并且不能根据用户的请求或热门话题而改变策略，从而导致更多缓存未能命中。此外，基准算法独立处理每个内容，不试图从过去的经验中学习流行度和上下文信息之间的关系。例如，当内容从缓存节点中淘汰时，与此内容相关的所有知识都将丢失，无法用于将来的决策。相反，基于流行度预测的缓存算法可以不断学习，并将学习到的知识存储到学习数据库中，以便将来使用。当缓存容量较小时，本算法的优势更大，因为需要做出更仔细的缓存决策。为了说明采用本算法在减少缓存需求方面的巨大优化，本文考虑 50% 的通用目标缓存命中率。在这种情况下，基于聚类的内容缓存（Cluster Based Content Caching, CBCC）需要 5 000 MB 的缓存容量，而 LRU 需要 9 000 MB 的缓存容量。

缓存命中率与请求时间的关系如图 5 所示，可显示缓存性能随请求时间的变化情况。图 5 中曲线上的每一点表示当前时间窗口和下一时间窗口内缓存命中的百分比。有以下几点值得注意。

（1）第一天之后，基于流行度预测的缓存算法优势随着更多请求的到来而变得明显。在这段时间里，本算法成功地从大量的请求中学习，从而做出准确的流行度

预测。

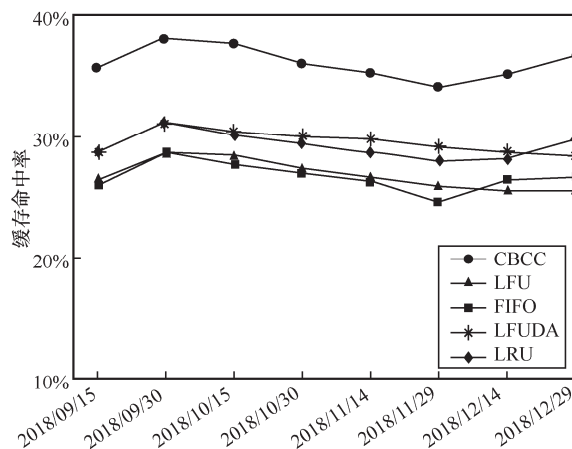
（2）如图 5 所示，本算法缓存命中率始终高于其他算法，但并不总是在提高。这是由于本文使用的数据集不足以支持大量独立的请求，曲线是请求到达单个缓存节点场景下实现的。当对大量缓存节点进行平均时，缓存命中率预计不会降低。尽管如此，图 4 和图 5 仍然可以说明本算法的学习行为及其优于现有解决方案的性能。



图注： ϕ 为 1.6

图 4 不同缓存容量下的缓存命中率

（3）LFU 和 LFUDA 未能跟踪流行内容的变化趋势，两种算法的缓存命中率在几十天后迅速下降。这是因为 LFU 使用过去流行的内容进行缓存决策，随着时间的推移，这些内容会过时。LFUDA 通过引入缓存年龄计数器来缓解这个问题，但并没有完全消除它。相比之下，本算法对流行程度分布的变化反应迅速，因此保持了稳定的缓存命中率。



图注：缓存容量为 2 500 MB， ϕ 为 1.6

图 5 缓存命中率与请求时间的关系

5 结束语

本文提出了一种在线学习方法来执行高效的缓存替换,所提算法通过聚类预测内容的流行度做出缓存替换决策。不同内容的访问模式存在相似性,因此可以利用聚类的方法对内容随时间动态分类,进而预测内容的流行度,减少缓存节点访问核心网云服务器的次数,最终改善用户体验。本文将本算法与现有的几种缓存策略进行比较,验证了其可行性和有效性。

参考文献:

- [1] PANWAR N, SHARMA S, SINGH A K. A survey on 5G: the next generation of mobile communication[J]. *Physical Communication*, 2016, 18: 64-84.
- [2] OBIKE W. Ericsson mobility report[R]. 2016.
- [3] DUAN T, DINAVAH V. Starlink space network-enhanced cyber-physical power system[J]. *IEEE Transactions on Smart Grid*, 2021, 12(4): 3673-3675.
- [4] CHIEN W C, LAI C F, HOSSAIN M S, et al. Heterogeneous space and terrestrial integrated networks for IoT: architecture and challenges[J]. *IEEE Network*, 2019, 33(1): 15-21.
- [5] QIU Y, NIU J W, ZHU X Z, et al. Mobile edge computing in space-air-ground integrated networks: architectures, key technologies and challenges[J]. *Journal of Sensor and Actuator Networks*, 2022, 11(4): 57.
- [6] RAY P P. A review on 6G for space-air-ground integrated network: key enablers, open challenges, and future direction[J]. *Journal of King Saud University - Computer and Information Sciences*, 2022, 34(9): 6949-6976.
- [7] ALI W, SHAMSUDDIN S M, ISMAIL A S. Intelligent Web proxy caching approaches based on machine learning techniques[J]. *Decision Support Systems*, 2012, 53(3): 565-579.
- [8] BERNARDINI C, SILVERSTON T, FESTOR O. A comparison of caching strategies for content centric networking[C]//*Proceedings of the 2015 IEEE Global Communications Conference (GLOBECOM)*. Piscataway: IEEE Press, 2015: 1-6.
- [9] BORST S, GUPTA V, WALID A. Distributed caching algorithms for content distribution networks[C]//*2010 Proceedings IEEE INFOCOM*. Piscataway: IEEE Press, 2010: 1-9.
- [10] DEHGHAN M, SEETHARAM A, JIANG B, et al. On the complexity of optimal routing and content caching in heterogeneous networks[C]//*Proceedings of the 2015 IEEE Conference on Computer Communications (INFOCOM)*. Piscataway: IEEE Press, 2015: 936-944.
- [11] DOBRIAN F, SEKAR V, AWAN A, et al. Understanding the impact of video quality on user engagement[J]. *ACM SIGCOMM Computer Communication Review*, 2011, 41(4): 362-373.
- [12] ZHANG P Y, LI Y J, KUMAR N, et al. Distributed deep reinforcement learning assisted resource allocation algorithm for space-air-ground integrated networks[J]. *IEEE Transactions on Network and Service Management*, 2023, 20(3): 3348-3358.
- [13] GREGORI M, GÓMEZ-VILARDEBÓ J, MATAMOROS J, et al. Wireless content caching for small cell and D2D networks[J]. *IEEE Journal on Selected Areas in Communications*, 2016, 34(5): 1222-1234.
- [14] JIN S D, BESTAVROS A. Popularity-aware greedy dual-size Web proxy caching algorithms[C]//*Proceedings of the 20th IEEE International Conference on Distributed Computing Systems*. Piscataway: IEEE Press. 2002: 254-261.
- [15] LI S H, XU J, VAN DER SCHAAR M, et al. Popularity-driven content caching[C]//*Proceedings of the IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*. Piscataway: IEEE Press, 2016: 1-9.
- [16] LI C L, ZHANG Y, SUN Q Q, et al. Collaborative caching strategy based on optimization of latency and energy consumption in MEC[J]. *Knowledge-Based Systems*, 2021, 233: 107523.
- [17] POULARAKIS K, IOSIFIDIS G, TASSIULAS L. Approximation algorithms for mobile data caching in small cell networks[J]. *IEEE Transactions on Communications*, 2014, 62(10): 3665-3677.
- [18] LYU F, YANG P, WU H Q, et al. Service-oriented dynamic resource slicing and optimization for space-air-ground integrated vehicular networks[J]. *IEEE Transactions on Intelligent Transportation Systems*, 2022, 23(7): 7469-7483.
- [19] SCELLATO S, MASCOLO C, MUSOLESI M, et al. Track globally, deliver locally: improving content delivery networks by tracking geographic social cascades[C]//*Proceedings of the 20th International Conference on World Wide Web*. New York: ACM, 2011: 457-466.

- [20] SERMPEZIS P, GIANNAKAS T, SPYROPOULOS T, et al. Soft cache hits: improving performance through recommendation and delivery of related content[J]. IEEE Journal on Selected Areas in Communications, 2018, 36(6): 1300-1313.
- [21] XU Q C, SU Z, FANG D F, et al. Hierarchical bandwidth allocation for social community-oriented multicast in space-air-ground integrated networks[J]. IEEE Transactions on Wireless Communications, 2023, 22(3): 1915-1930.
- [22] SHANMUGAM K, GOLREZAEI N, DIMAKIA A, et al. Femto-caching: wireless video content delivery through distributed caching helpers[J]. IEEE Transactions on Information Theory, 2013, 59(12): 8402-8413.
- [23] SHUJA J, BILAL K, ALASMARY W, et al. Applying machine learning techniques for caching in next-generation edge networks: a comprehensive survey[J]. Journal of Network and Computer Applications, 2021, 181: 103005.
- [24] SITARAMAN R K, KASBEKAR M, LICHTENSTEIN W, et al. Overlay networks: an akamai perspective[J]. Advanced Content Delivery, Streaming, and Cloud Services, 2014: 305-328.
- [25] WANG X F, WANG C Y, LI X H, et al. Federated deep reinforcement learning for Internet of things with decentralized cooperative edge caching[J]. IEEE Internet of Things Journal, 2020, 7(10): 9441-9455.
- [26] YUAN P Y, SHAO S K, GENG L J, et al. Caching hit ratio maximization in mobile edge computing with node cooperation[J]. Computer Networks, 2021, 200: 108507.
- [27] ZHAO Y W, LI R B, WANG C Y, et al. Neighboring-aware caching in heterogeneous edge networks by actor-attention-critic learning[C]//Proceedings of the ICC 2021 - IEEE International Conference on Communications. Piscataway: IEEE Press, 2021: 1-6.

[作者简介]



仇超 (1988—), 女, 博士, 天津大学副教授, 主要研究方向为算力网络、边缘计算、边缘智能。



王晨阳 (1989—), 男, 博士, 深圳大学计算机与软件学院博士后, 主要研究方向为边缘计算、大数据分析、面向大模型的边缘智能。