

# 基于“推荐-学习”的两阶段数据布局策略

梁 杨<sup>1,2</sup>, 丁长松<sup>1,2</sup>, 胡志刚<sup>3</sup>

(1.湖南中医药大学信息科学与工程学院,湖南 长沙 410208)

(2.湖南省中医药大数据分析实验室,湖南 长沙 410208)

(3.中南大学计算机学院,湖南 长沙 410083)

[摘要] 针对数据布局不合理导致云边协同集群服务质量下降和运营开销增加等问题,提出一种基于“推荐-学习”的两阶段数据副本管理机制.鉴于数据密集型移动应用的特点,综合考虑了云边环境下的数据访问延迟和放置代价之间的最优权衡.从理论上构建了副本放置的多目标数学模型,将决策问题描述为具有延迟和成本约束的双目标优化问题;在“推荐”阶段,设计了一个基于移动预测和反馈优化的副本推荐引擎,减少了副本创建的盲目性;在“学习”阶段,构建了一个基于异步优势行动者-评论家算法(A3C)的强化学习副本放置规则学习模型,改进了副本服务的全局性能指标.实验结果表明,基于“推荐-学习”的两阶段数据布局策略能够有效地减少等待延迟和节约成本开销,为现代云边协同系统的数据管理服务提供行之有效的解决方案,具有重要的理论意义和应用价值.

[关键词] 数据布局,服务质量,数据密集型,副本推荐,规则学习

[中图分类号]TP391 [文献标志码]A [文章编号]1001-4616(2023)04-0080-11

## A Two-Stage Data Placement Strategy Based on Recommendation and Learning

Liang Yang<sup>1,2</sup>, Ding Changsong<sup>1,2</sup>, Hu Zhigang<sup>3</sup>

(1.School of Informatics, Hunan University of Chinese Medicine, Changsha 410208, China)

(2.Big Data Analysis Laboratory of TCM in Hunan Province, Changsha 410208, China)

(3.School of Computer Science and Engineering, Central South University, Changsha 410083, China)

**Abstract:** Aiming at the problem that the unreasonable data layout leads to the decrease of service quality and the increase of operation cost in cloud-edge collaborative cluster, a two-stage data replica management mechanism based on ‘recommendation and learning’ method is proposed. In view of the characteristics of data-intensive mobile applications, the optimal trade-off between data access latency and placement cost in the cloud environment is considered. The multi-objective mathematical model of replica placement is established in theory, and the decision problem is described as a bi-objective optimization problem with delay and cost constraints. In the recommendation stage, a replica recommendation engine based on motion prediction and feedback optimization is proposed to reduce the blindness of replica creation. In the learning stage, a reinforcement learning replica placement rule learning model based on Asynchronous Advantage Actor-Critic algorithm(A3C) is constructed to improve the global performance index of replica service. The experimental results show that the two-stage data placement strategy based on ‘recommendation and learning’ method can reduce latency time and save cost overhead effectively and efficiently. Meanwhile, our proposed method provides an effective solution for challenges of data management in the real-life cloud-edge collaboration systems, which has important theoretical significance and application value.

**Key words:** data placement, quality of service, data-intensive, replica recommendation, rule learning

收稿日期:2023-05-30.

基金项目:国家自然科学基金项目(62172442)、湖南省教育厅优秀青年项目(22B0400)、湖南中医药大学校级科研基金(2021XJJ021)、湖南省自然科学基金项目(2023JJ60124)、长沙市科技局项目(kq2202265)、湖南中医药大学学科建设“揭榜挂帅”项目(22JBZ049).

通讯作者:胡志刚,博士,教授,博士生导师,研究方向:高性能计算、云计算、边缘计算和能效资源管理. E-mail:zgghu@csu.edu.cn

近年来,智能手机、联网汽车、AR/VR 等智能设备快速普及,各类移动终端产生的数据量呈爆炸式增长<sup>[1]</sup>. 多接入边缘计算(multi-access edge computing, MEC)作为一种邻近计算范式,其本质是云计算向边缘网络的延伸. MEC 允许边缘服务器分布在不同的地理位置,将计算和存储资源部署在移动用户附近,因此可以提供比云计算服务更低延迟<sup>[2]</sup>. MEC 框架在满足访问时延的前提下,在节省网络带宽、提高服务质量(quality of service, QoS)、缓解数据中心压力等方面具有天然的优势. 然而,在数据密集型应用场景下因资源受限所带来的一系列问题正引起研究者的广泛关注<sup>[3]</sup>. 由于边缘节点存储资源有限,数据密集型应用会频繁地进行远程数据访问,从而导致较高的往返延迟,甚至完全抵消 MEC 的优势<sup>[4]</sup>.

在 MEC 环境中,大量移动边缘设备将不断产生和消费各类数据,不恰当的副本管理策略极易导致网络资源浪费和通信延迟过高<sup>[5]</sup>. 一方面,当处理本地用户任务的数据访问请求时,由于边缘节点存在严重自治,通常需要分别创建副本以减少远程数据访问的次数,很容易出现副本的高冗余存储和高频次更新,导致“副本泛滥”、资源浪费等问题<sup>[6]</sup>;另一方面,云边网络用户对数据副本的多边高并发请求具有随机性和区域性,增加了“副本管理失控”的风险<sup>[7]</sup>. 因此对数据副本进行全局部署时,需要综合权衡局部和全局的资源互补与冲突,进而确定最佳副本布局.

一般,不合理的副本部署不仅会增加任务响应延迟,还会增加网络和存储开销<sup>[8]</sup>. 为解决上述问题,本文提出一种基于“推荐-学习”的两阶段数据布局策略(two-stage data placement strategy, TDPS),通过“推荐”阶段和“学习”阶段对副本放置规则进行深度优化. TDPS 综合考虑了副本流行度和用户移动性,旨在实现访问延迟和成本开销的权衡优化. 综上所述,本文的主要贡献包括 4 个方面:

(1)提出了一种基于“推荐-学习”的两阶段副本管理框架,通过将 MEC 环境下副本决策过程分解为“边缘推荐”和“整体学习”两个阶段,有效整合了边缘端的局部信息和云端的全局信息,优化了副本管理机制;

(2)对目标问题进行形式化描述并构建了数学模型,将 MEC 中的副本放置决策问题描述为具有延迟和成本约束的双目标优化问题;

(3)在“推荐”阶段,提出一个基于移动预测和反馈优化的副本推荐引擎,一定程度上解决了盲目创建副本的问题;在“学习”阶段,提出一个基于 A3C 强化学习的副本放置规则学习模型,从全局视角下进一步保证了云边系统的数据服务性能最优;

(4)实验结果表明,本文基于“推荐-学习”的两阶段数据布局策略可以有效降低数据管理成本并改善用户 QoS 满意度.

## 1 相关工作

随着云计算和边缘计算逐步融合,如何提高云边系统的副本管理效率成为一个新兴的研究热点,引起国内外研究者的广泛关注<sup>[9]</sup>. 一些研究学者提出了基于数据流行度的静态副本策略<sup>[10]</sup>,在指定的存储节点上设置固定的副本数量;而另外一些研究人员则提出了动态副本策略<sup>[11-12]</sup>,允许根据用户的行为特征动态地调整副本数量和位置. Jin 等<sup>[13]</sup>提出了一种边端协作存储框架,在边缘计算环境中协作存储必要的的数据,以提高边缘层的性能,减少将任务转移到云数据中心的可能性. Vales 等<sup>[14]</sup>提出一个将移动设备的存储资源和雾节点集中起来的解决方案,有效解决了用户访问远程云时面临的访问延迟、带宽受限、通信开销和位置盲目性等问题. Chang 等<sup>[15]</sup>通过分析边缘服务器的数据副本收入、成本和利润,提出了面向 MEC 的自适应副本机制,有效地缩短了任务平均响应时间,提高了数据分组的服务质量.

由于边缘端存储资源和通信带宽都存在一定限制,副本优化机制需要充分考虑资源受限条件下的副本管理. Teranishi 等<sup>[16]</sup>提出了一种面向边缘环境下物联网应用的动态数据流平台,不仅能够使物联网应用实现较小的延迟,而且可以避免物联网应用对网络和计算资源造成过载;研究者还提出了一个具有多因素最少使用算法的分布式多级存储模型<sup>[17]</sup>,为处理海量物联网数据时可能面临的存储受限和网络拥塞等问题提供了有效的解决方案;Aral 等<sup>[18]</sup>提出了一种智能分布式数据部署方案,解决了数据访问延迟和副本放置成本等问题.

以上早期研究虽然考虑了边缘端资源的有限性,但主要不足之处在于没有充分利用云端已有的高性能资源. 对此,一些研究者针对云边系统的副本管理优化提出了若干新方法,其中最为突出的是基于机器

学习的副本管理. Wang 等<sup>[19]</sup>提出了一种基于强化学习的副本缓存框架,用于激励边缘设备为周边用户提供副本存储服务;为了能够自适应地利用网络资源并降低任务响应时间,研究者还提出了一种基于深度强化学习的智能资源分配方案<sup>[20]</sup>. 以上研究显示,机器学习技术在实现“在线副本服务”方面具有较好的自适应性和鲁棒性.

综上所述,优化数据布局既需要考虑边缘资源的受限性,还需要考虑网络传输速度的差异性. 本文从以上两个方面来分析现有系统数据服务的性能瓶颈,并提出基于“推荐-学习”的两阶段数据布局策略来动态协调云端和边缘端的存储资源分配,以期充分利用云端的高性能来弥补边缘端的资源短缺.

2 系统框架和数学模型

根据云边环境下面向移动用户的典型应用场景,本节提出了一个两阶段副本管理框架,并对目标问题进行模型构建.

2.1 基于“推荐-学习”的两阶段副本管理框架

在云边协同系统中,一个有效的数据布局策略必须充分考虑云端和边缘端资源性能的差异. 针对由此导致的一系列性能瓶颈问题,本文提出了一种基于“推荐-学习”的两阶段副本管理机制 TDPS,其整体架构如图 1 所示.

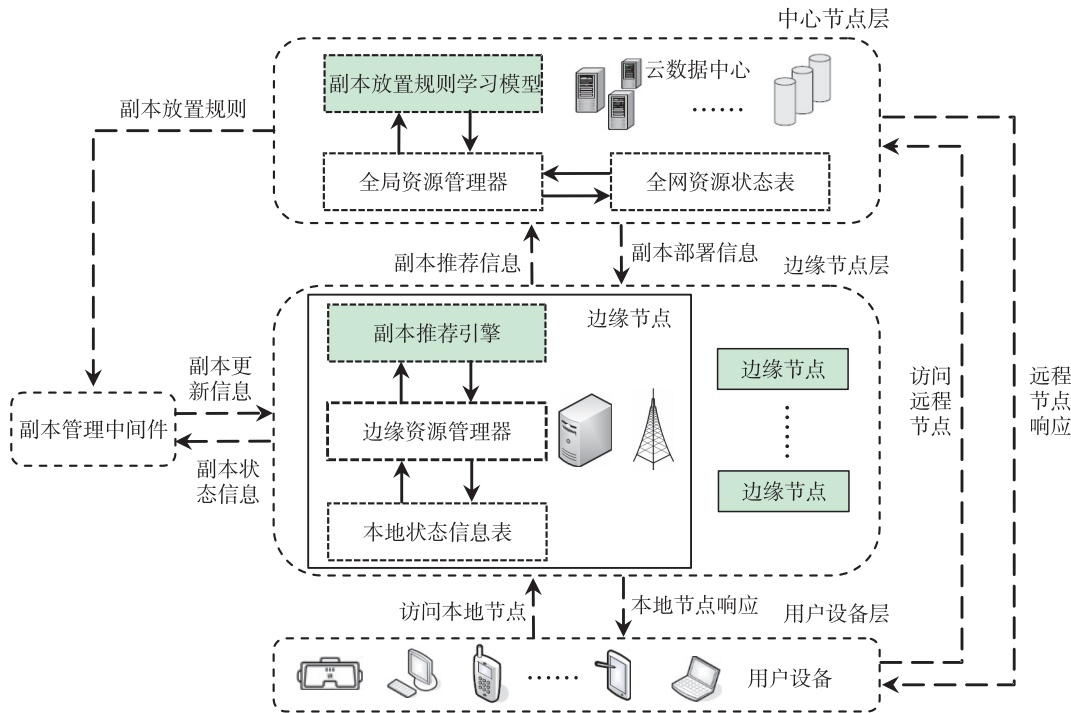


图 1 基于“推荐-学习”的两阶段副本管理架构  
Fig. 1 The system architecture of TDPS

- 在图 1 中,TDPS 架构从逻辑上自下而上主要包含 3 层:用户设备层、边缘节点层和中心节点层.
- (1) 用户设备层: 主要包含各种具有移动性的用户设备,需要向边缘云或中心云请求数据服务.
  - (2) 边缘节点层: 主要由分布式边缘节点组成,各边缘节点部署有副本推荐引擎、边缘资源管理器和本地状态信息表,负责向中心节点层推荐热点副本.
  - (3) 中心节点层: 云数据中心是本层的重要组件,主要包括副本放置规则学习模型、全局资源管理器和全网资源状态表,负责对全网数据布局进行决策.

2.2 问题描述与系统建模

假设  $E = \{e_1, e_2, \dots, e_n\}$  表示所有边缘节点的集合,  $n$  为边缘节点的总数;  $F = \{f_1, f_2, \dots, f_m\}$  表示所有可访问的文件全集,  $m$  为文件总数;  $SZ_{f_i}$  表示文件  $f_i$  的大小. 用户所感知的访问延迟主要依赖于用户和待访问文件之间的距离,则用户和其对应边缘节点之间的平均等待延迟如式(1)所示.

$$ALE = \frac{\sum_{j=1}^n \sum_{i=1}^m (NR_{f_i} \cdot SZ_{f_i} \cdot X_{f_i, e_j})}{VE \sum_{j=1}^n \sum_{i=1}^m (NR_{f_i} \cdot X_{f_i, e_j})}, \quad (1)$$

式中,  $VE$  表示边缘节点的数据传输速率;  $NR_{f_i}$  表示用户请求访问文件  $f_i$  的总次数.  $X_{f_i, e_j}$  是一个二进制变量, 用以表示  $f_i$  的副本是否位于  $e_j$  上, 如果  $f_i$  的副本位于  $e_j$  上, 则  $X_{f_i, e_j} = 1$ ; 否则  $X_{f_i, e_j} = 0$ .

简化起见, 假设云数据中心存储了所有文件的副本, 如果用户对  $f_i$  的访问请求没有在本地得到满足, 则必须向远程云请求  $f_i$ , 用户与云数据中心之间的平均等待延迟可以表示为式(2).

$$ALC = \frac{\sum_{j=1}^n \sum_{i=1}^m (NR_{f_i} \cdot SZ_{f_i} \cdot \bar{X}_{f_i, e_j})}{VC \cdot \sum_{j=1}^n \sum_{i=1}^m (NR_{f_i} \cdot \bar{X}_{f_i, e_j})}, \quad (2)$$

式中,  $VC$  表示云数据中心的数据传输速率.  $\bar{X}_{f_i, e_j}$  表示对  $X_{f_i, e_j}$  取反后的一个二进制变量, 如果边缘节点  $e_j$  上没有存储文件  $f_i$  的副本, 则  $\bar{X}_{f_i, e_j} = 1$ , 用户将从云数据中心获取文件  $f_i$ ; 否则  $\bar{X}_{f_i, e_j} = 0$ .

此外, 资源开销同样制约着数据布局问题的决策. 本文主要考虑副本放置过程中网络传输成本和边缘存储成本对总成本的影响. 网络传输成本是指文件从云数据中心传输到边缘节点所产生的网络传输开销, 如式(3)所示.

$$C_{tx} = \sum_{j=1}^n \sum_{i=1}^m T_{unit} \cdot SZ_{f_i} \cdot X_{f_i, e_j}, \quad (3)$$

式中,  $T_{unit}$  表示从云数据中心到边缘节点的单位数据传输成本,  $C_{tx}$  表示总传输开销.

此外, 因副本放置造成的存储开销亦不容忽视, 则边缘节点的副本存储成本可以表示为式(4)所示.

$$C_{sto} = \sum_{j=1}^n \sum_{i=1}^m S_{unit} \cdot SZ_{f_i} \cdot X_{f_i, e_j}, \quad (4)$$

式中,  $S_{unit}$  表示边缘节点上单位数据的存储成本,  $C_{sto}$  表示总存储开销.

目标问题进一步转化为式(5)所示.

$$\Psi = \mu(ALE + ALC) + \eta(C_{tx} + C_{sto}), \quad (5)$$

式中,  $\mu + \eta = 1, 0 < \mu, \eta < 1$ .

基于上述分析, 为表示延迟和成本敏感的数据布局问题, 构建了如式(6)所示的双目标优化函数.

$$\begin{aligned} \min_{X_{f_i, e_j}} \Psi \\ \text{s.t.} \quad \sum_{i=1}^m S_{unit} \cdot SZ_{f_i} \cdot X_{f_i, e_j} \leq S_{e_j}^{avl}, e_j \in E \end{aligned} \quad (6)$$

式中,  $S_{e_j}^{avl}$  表示边缘节点  $e_j$  上的可用存储容量, 约束条件表示放置在边缘节点上的副本总大小不能超过该节点的可用存储容量.

### 3 基于“推荐-学习”的两阶段副本管理机制

本节提出一种基于“推荐-学习”的两阶段副本管理机制, 综合权衡全网的 QoS 和资源开销, 进而确定当前工作周期内最优的全局副本放置规则. 在副本推荐阶段, 通过在边缘节点上构建副本推荐引擎快速、准确地量化边缘层副本访问价值, 缩小存在副本需求的局部文件规模, 避免云端/边缘端盲目创建副本, 使服务提供商在保证 QoS 的同时减少不必要的资源开销; 在副本放置阶段, 云数据中心基于强化学习构建全局副本放置规则学习模型, 优化数据布局, 以期指导副本管理中间件进行副本放置和替换, 在 QoS 和成本开销间达到最佳折中, 从整体上改进云边环境下副本服务的各类性能指标.

#### 3.1 基于移动预测和反馈优化的副本推荐引擎

为了快速、准确地量化边缘端副本价值, 最大程度减小副本规模, 本文首先在边缘节点上构建副本推荐引擎, 从局部层面上避免副本的盲目创建.

一方面, 推荐引擎必须充分考虑用户移动性和文件访问特性, 从而准确评估文件访问概率; 另一方面,

引入反馈机制及时修正误差,提高推荐引擎自适应性. 基于此,副本推荐引擎的框架如图 2 所示.

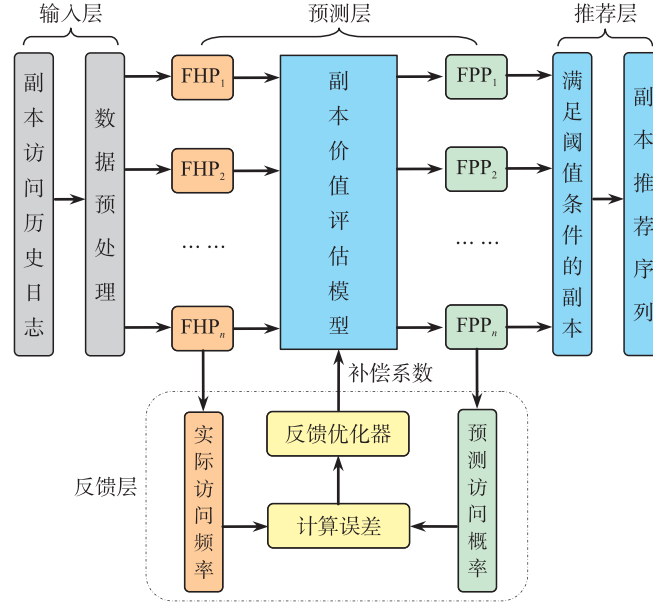


图 2 基于移动预测和反馈优化的副本推荐引擎

Fig. 2 Replica recommendation engine based on motion prediction and feedback optimization

由图 2 可以看出,副本推荐引擎主要由输入层、预测层、反馈层和推荐层组成. 其中,输入层负责对数据进行预处理以满足模型输入的要求;预测层根据文件历史流行度和用户移动性来预测本地节点上不同副本的期望访问概率;反馈层负责将副本的预测误差输入到反馈优化器,计算误差期望并确定补偿系数,及时反馈给预测层优化预测精度;推荐层负责筛选满足条件的副本,从中选择价值排名靠前的若干副本,构成副本推荐序列.

假设  $U = \{u_1, u_2, \dots, u_z\}$  表示所有用户的集合,  $z$  表示用户的总数;  $T = \{t_1, t_2, \dots, t_w\}$  表示所有时段的集合,  $w$  表示时段的总数;  $FHP_{f_i, e_j}^{t_h}$  表示在时段  $t_h$  内  $f_i$  在  $e_j$  上的历史流行度;  $P_{u_k, e_j}^{t_c}$  表示在当前时段  $t_c$  内用户  $u_k$  处于  $e_j$  服务范围内的概率. 另外,已有研究表明采用马尔可夫过程来描述用户移动性是合理的<sup>[21]</sup>. 根据马尔可夫过程的性质,用户在下一时段出现在  $e_s$  上的概率可以表示为式(7).

$$P_{u_k, e_s}^{t_{c+1}} = P_{u_k, e_j}^{t_c} \cdot P_{u_k, e_j \rightarrow e_s}^{t_{c+1}}, \quad (7)$$

式中,  $P_{u_k, e_j \rightarrow e_s}^{t_{c+1}}$  表示在下一个时段  $t_{c+1}$  内用户  $u_k$  从  $e_j$  移动到  $e_s$  的概率,  $P_{u_k, e_s}^{t_{c+1}}$  表示在  $t_{c+1}$  内用户  $u_k$  处于  $e_s$  服务范围内的概率.

假设  $PR_{u_k, f_i}^{t_{c+1}}$  表示在  $t_{c+1}$  内用户  $u_k$  请求访问  $f_i$  的概率,其服从泊松分布;  $FPP_{f_i, e_s}^{t_{c+1}}$  表示在  $t_{c+1}$  内  $f_i$  在  $e_s$  上的期望访问概率. 在此基础上,副本价值评估模型如式(8)所示.

$$FPP_{f_i, e_s}^{t_{c+1}} = FHP_{f_i, e_s}^{t_c} \sum_{k=1}^z P_{u_k, e_s}^{t_{c+1}} \cdot PR_{u_k, f_i}^{t_{c+1}}, \quad (8)$$

由此可见,副本推荐引擎同时考虑了边缘节点上的文件历史访问流行度和用户移动性的影响.

为改善价值评估模型性能,提高模型自适应能力,本文在推荐引擎中引入反馈层,对预测层结果进行误差分析,通过补偿系数进一步对模型进行修正,则式(8)进一步改进为式(9)所示.

$$FPP_{f_i, e_s}^{t_{c+1}} = \lambda^{t_c} FHP_{f_i, e_s}^{t_c} \sum_{k=1}^z P_{u_k, e_s}^{t_{c+1}} \cdot PR_{u_k, f_i}^{t_{c+1}}, \quad (9)$$

式中,  $\lambda^{t_c}$  表示在时段  $t_c$  时的模型补偿系数,且  $\lambda^{t_c} > 0$ .

最后,推荐引擎将根据副本期望访问概率进行推荐,具体推荐过程如算法 1 所示.

首先,将  $t_c$  时段边缘节点  $e_s$  上的所有文件的期望访问概率集作为推荐层的输入;其次,根据价值阈值筛选出期望访问概率不低于该阈值的文件;最后,如果存在满足阈值要求的文件,则返回期望访问概率中排名前  $K$  ( $K > 0$ ) 的副本标识符列表.



## 算法 1 副本推荐算法

输入: 在  $t_{c+1}$  时段每个文件在边缘节点  $e_s$  上的期望访问概率  $FPP_{f_i, e_s}^{t_{c+1}}$ ;

输出: 在  $t_c$  时段边缘节点  $e_s$  上的高价值副本序列  $RFS_{e_s}^t[\cdot]$ ;

```

① if  $FPP_{f_i, e_s}^{t_{c+1}} \neq \text{NULL}$  then
②   for  $i = 1$  to  $m$  do
③     if  $FPP_{f_i, e_s}^{t_{c+1}} \geq FPP_{\text{threshold}}$  then
④        $FPPNew[i] = FPP_{f_i, e_s}^{t_{c+1}}$ ;
⑤     else
⑥        $FPPNew[i] = 0$ ;
⑦     end if
⑧   end for
⑨   if  $FPPNew \neq \text{NULL}$  then
⑩      $FPPTopK[\cdot] = \text{TopK}(FPPNew)$ ;
⑪      $RFS_{e_s}^t[\cdot] = \text{GetFileID}(FPPTopK)$ ;
⑫   else
⑬      $RFS_{e_s}^t[\cdot] = \text{NULL}$ ;
⑭   end if
⑮ else
⑯    $RFS_{e_s}^t[\cdot] = \text{NULL}$ ;
⑰ end if
⑱ return  $RFS_{e_s}^t[\cdot]$ 

```

## 3.2 基于 A3C 强化学习的副本放置规则学习模型

边缘端推荐的副本序列存在一定的局部性限制,为保证全局层面的 QoS 和系统性能最优,本文提出一种基于异步优势行动者-评论家算法(asynchronous advantage actor-critic, A3C)的副本放置规则学习模型,采用基于多线程机制的异步训练框架,引入优势函数对策略网络(Actor)和价值网络(Critic)进行更新,使模型在训练速度、收敛性能和预测精度等方面具有更好的表现. 该模型由多个并行子线程和一个全局网络组成,每个子线程由一个 Actor 和一个 Critic 组成,负责独立运行 Actor-Critic 算法和并行进行参数探索,既独立更新全局网络参数,又从全局网络获取参数指导;全局网络主要包括一个全局 Actor 和一个全局 Critic,负责汇总子线程与环境交互的结果,异步更新全局网络参数.

设环境的状态空间为  $S = \{S_1, S_2, \dots, S_T\}$ ,  $S_t$  表示  $t$  时刻的状态;  $A_t = \{A_t^1, A_t^2, \dots, A_t^n\}$  为  $t$  时刻选择的动作,其中  $A_t^j = \{a_1, a_2, \dots, a_m\}$  表示  $t$  时刻在  $e_j$  上执行的动作,  $a_i \in \{0, 1\}$ , 若在该节点上创建  $f_i$  的副本则  $a_i = 1$ , 否则  $a_i = 0$ ; 策略  $\pi$  表示副本放置在边缘节点上的概率分布,如式(10)所示.

$$\pi = \begin{bmatrix} \pi_{1,1} & \pi_{1,2} & \cdots & \pi_{1,m} \\ \pi_{2,1} & \pi_{2,2} & \cdots & \pi_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ \pi_{n,1} & \pi_{n,2} & \cdots & \pi_{n,m} \end{bmatrix} = [\pi_1, \pi_2, \dots, \pi_n]^T, \quad (10)$$

式中,  $T$  表示矩阵转置,  $\pi_j$  表示  $e_j$  上的副本放置模式,  $\pi_{j,i}$  表示  $f_i$  的副本放置在  $e_j$  上的概率,  $\pi_{j,i} \in [0, 1]$ .  $\pi_{j,i} = 0$  表示  $f_i$  的副本不能放置在  $e_j$  上;反之,  $\pi_{j,i} = 1$ .

为有效度量智能体所选动作策略的优劣,本文提出一个延迟-成本组合奖励函数,如式(11)所示.

$$LC_t(S_t, A_t) = \frac{1}{L_t(S_t, A_t) \cdot C_t(S_t, A_t)}, \quad (11)$$

式中,  $L_t(S_t, A_t)$  和  $C_t(S_t, A_t)$  分别表示在状态-动作对  $(S_t, A_t)$  下的访问延迟和成本开销,  $LC_t(S_t, A_t)$  表示组合奖励. 基于此,长期累计奖励如式(12)所示.

$$\mathcal{R}_t = \sum_{k=0}^{\infty} \gamma^k LC_{t+k}(S_{t+k}, A_{t+k}), \quad (12)$$

式中,  $\gamma \in (0, 1]$  为折扣因子,表示从环境状态  $S_t$  开始,未来环境奖励对累计奖励的影响权重. 显然,模型训练目标可以表示为最大化长期累计奖励的期望,如式(13)所示.

$$\max_{\pi} E_{S \sim F_0} [\mathcal{R}_t | S_t = S], \quad (13)$$

式中,环境状态  $S$  服从先验分布  $F_0$ ,  $\pi$  表示在环境状态  $S_t$  下选择的副本放置策略,则此时状态价值函数如式(14)所示.

$$V^{\pi}(S_t) = E_{\pi} [\mathcal{R}_t | S_t = S], \quad (14)$$

类似地,智能体的动作价值函数如式(15)所示.

$$Q^{\pi}(S_t, A_t) = LC_t(S_t, A_t) + \gamma V^{\pi}(S_{t+1}), \quad (15)$$

为寻找使长期累计奖励期望最大化的副本放置策略  $\pi$ ,需要不断更新策略参数  $\theta$ ,对此,智能体采用优势函数来评价某一策略动作所产生的增益大小,如式(16)所示.

$$A^{\pi}(S_t, A_t) = Q^{\pi}(S_t, A_t) - V^{\pi}(S_t), \quad (16)$$

式中,  $V^{\pi}(S)$  的值由 Critic 网络训练获得,若  $A^{\pi}(S, A) > 0$ ,则表示在状态  $S_t$  下执行的动作有利于奖励的增加,策略参数可以朝梯度方向更新以改进副本放置策略;反之,则不建议更新策略参数. 为加快长期累计奖励期望收敛,通过梯度上升法更新 Actor 策略网络参数  $\theta$ ,如式(17)所示.

$$\theta = \theta + \mu \nabla_{\theta} \log \pi(A_t | S_t; \theta) A^{\pi}(S_t, A_t), \quad (17)$$

式中,  $\pi(A_t | S_t; \theta) \in [0, 1]$  表示在  $S_t$  和  $\theta$  条件下执行  $A_t$  的概率,  $\log \pi(A_t | S_t; \theta) \in [0, +\infty)$  避免了梯度消失.  $\mu$  为学习率,  $\mu \in [10e-6, 1]$ . 另外,为避免过早陷入局部收敛,在策略函数的损失函数中需增加  $\pi$  的交叉熵项,则式(17)可改进为式(18)所示.

$$\theta = \theta + \mu \nabla_{\theta} \log \pi(A_t | S_t; \theta) A^{\pi}(S_t, A_t) + \eta \nabla_{\theta} H[\pi(S_t; \theta)], \quad (18)$$

式中,  $\eta$  为交叉熵权重因子,  $\eta \in [0, 1]$ . 基于时间差分方法进行 Critic 价值网络参数更新,其价值函数的损失函数如式(19)所示.

$$\text{LOSS} = (Q^{\pi}(S_t, A_t) - V^{\pi}(S_t))^2, \quad (19)$$

为加快式(19)所示损失函数收敛,通过梯度下降法更新 Critic 策略网络参数  $\theta_v$ ,如式(20)所示.

$$\theta_v = \theta_v + \varepsilon \frac{\partial (Q^{\pi}(S_t, A_t) - V^{\pi}(S_t))^2}{\partial \theta_v}, \quad (20)$$

式中,  $\varepsilon$  为学习率,  $\varepsilon \in [10e-6, 1]$ . 对此,副本放置规则的 A3C 子线程学习算法伪代码如算法 2 所示.

#### 算法 2 A3C 子线程学习算法

输入:环境状态集合  $S$ ,智能体动作集合  $A$ ,A3C 全局网络的全局 Actor 参数  $\theta$  和全局 Critic 参数  $\theta_v$ ,A3C 子线程的 Actor 参数  $\theta'$  和 Critic 参数  $\theta'_v$ ,全局最大迭代次数  $T_{\max}$  和全局迭代计数器  $T$ ,子线程单次迭代最大时间序列  $t_{\max}$  和本地线程时间序列计数器  $t$ ,折扣因子  $\gamma$ ,学习率  $\mu, \varepsilon$ ,交叉熵权重因子  $\eta$ ;

输出:A3C 全局神经网络参数  $\theta, \theta_v$ .

- ①  $T=0$  and  $t=1$
- ② repeat
- ③  $d\theta \leftarrow 0$  and  $d\theta_v \leftarrow 0$ ;
- ④  $\theta' \leftarrow \theta$  and  $\theta'_v \leftarrow \theta_v$ ;
- ⑤  $t_{\text{start}} = t$  and  $S_t = \text{getState}(t)$ ;
- ⑥ repeat
- ⑦ 基于  $\pi(A_t | S_t; \theta')$  执行动作  $A_t$ ;
- ⑧ 获得  $LC_t(S_t, A_t)$  and  $S_{t+1}$ ;
- ⑨  $t \leftarrow t+1$  and  $T \leftarrow T+1$ ;
- ⑩ until  $t - t_{\text{start}} = t_{\max}$  or  $S_t = S_T$
- ⑪  $Q^{\pi}(S_t, A_t) = \begin{cases} 0 & S_t = S_T \\ V^{\pi}(S_t, \theta'_v) & S_t \neq S_T \end{cases}$ ;
- ⑫ for  $i=t-1$  to  $t_{\text{start}}$  do
- ⑬  $Q^{\pi}(S_i, A_i) = LC_i(S_i, A_i) + \gamma Q^{\pi}(S_{i+1}, A_{i+1})$ ;
- ⑭ 基于式(18)更新  $d\theta$ ;
- ⑮ 基于式(20)更新  $d\theta_v$ ;
- ⑯ end for
- ⑰  $\theta = \theta - \mu d\theta$  and  $\theta_v = \theta_v - \varepsilon d\theta_v$ ;
- ⑱ until  $T > T_{\max}$
- ⑲ return  $\theta$  and  $\theta_v$

## 4 结果与讨论

为评价和分析所提出算法的性能,本文引入副本命中率、平均访问延迟和成本节约率等<sup>[22]</sup>3项测试指标作为评估标准,将 TDPS 策略与其他基准算法进行比较,验证所提算法在副本决策方面的有效性。

### 4.1 实验环境设置

本文采用 EdgecloudSim<sup>[23]</sup>平台进行仿真实验,模拟云边环境. 边缘节点的位置分布由 BRITE<sup>[24]</sup>拓扑生成器给出,中心节点设置为距离各边缘较远的数据中心. 为了便于比较和分析,假定所有待访问文件的大小相等,用户提交的作业请求近似服从泊松分布,初始数据位置近似服从齐夫分布,用户优先向附近边缘节点提出访问请求,实验参数的详细设置如表 1 所示。

表 1 实验设置  
Table 1 Experimental settings

参数	定义	值
$c$	中心节点数量	1
$n$	边缘节点数量	20
$m$	原始数据文件个数	2 000
$z$	终端用户数	1 000
$B$	网络平均带宽(Mbps)	20
$SZ$	文件大小(MB)	64
$D_1$	用户与中心节点之间的数据平均加载延迟(ms)	200
$D_2$	用户与本地边缘节点间的数据平均加载延迟(ms)	5

### 4.2 实验结果与分析

本实验将 TDPS 分别与 D-ReP<sup>[18]</sup>、RPME<sup>[25]</sup>和 HRS<sup>[26]</sup>3种基准算法在副本命中率、平均访问延迟和成本节约率等方面进行对比,每次对比实验采用同一性能指标,分别进行 10 组独立重复实验,以评估 TDPS 策略的性能,具体实验结果与分析如下:

#### (1) 副本命中率

副本命中率(replica hit rate, RHR)表示本地副本访问次数占本地用户请求总数的比例,用以衡量该策略的本地化水平,如式(21)所示。

$$\text{RHR} = \frac{\sum_j^n \sum_i^m \text{Local} R_{f_i}^j}{\sum_j^n \sum_i^m i \text{Total} R_{f_i}^j}, \quad (21)$$

式中,  $\text{Total} R_{f_i}^j$  表示边缘节点  $j$  接收到用户请求访问文件  $f_i$  的总次数,  $\text{Local} R_{f_i}^j$  表示边缘节点  $j$  满足用户本地访问文件  $f_i$  的次数. 图 3 显示了在不同并发请求数下不同副本部署策略的副本命中率比较。

图 3 表明,当并发请求规模较小时,TDPS 的副本命中率与其他算法差距不大,甚至略低于某些算法;随着并发请求数的增加,副本命中率总体呈上升趋势,而 TDPS 的增长幅度明显高于其他算法,说明其具有更好的副本放置决策能力. 主要原因是 TDPS 同时考虑了文件历史流行度和用户移动性带来的影响,通过预测用户位置并提前放置数据副本,有效提高了副本命中率;当并发请求规模较大时,各算法副本命中率的变化幅度不明显,原因在于边缘节点存储空间有限,当副本数增加到一定程度后,仅进行必要的副本替换,副本命中率趋于稳定。

#### (2) 平均访问延迟

平均访问延迟(average access latency, AAL)表示从作业提交数据访问请求到作业执行完成的平均持续时间,如式(22)所示。

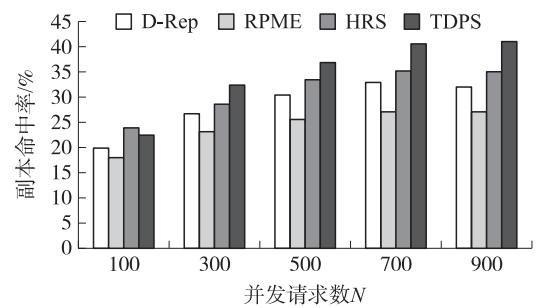


图 3 不同副本部署策略的副本命中率比较

Fig. 3 The comparison of replica hit ratio for various replica deployment strategies



$$AAL = \frac{\sum_{i=1}^Q (t_i(\text{end}) - t_i(\text{start}))}{Q}, \quad (22)$$

式中,  $t_i(\text{end})$  和  $t_i(\text{start})$  分别表示作业  $i$  的完成时间和开始时间,  $Q$  表示用户提交的作业总数. 图 4 对比了 TDPS 与基准算法在平均访问延迟方面的表现.

由图 4 可以看出, 当并发请求数较少时, 不同方法的平均访问延迟差别不大, TDPS 的平均访问延迟甚至高于 HRS, 这是因为不同策略创建的副本数量均处于较低水平, 使得服务响应时间较为接近; 当并发请求数达到 500 左右时, TDPS 的平均访问延迟比 D-ReP 减少约 17.9%, 比 RPME 减少约 24.6%, 原因在于 TRMM 具有更高的副本命中率; 随着并发请求的增加, TDPS 平均访问延迟的增长率明显低于其他算法, 进一步说明 TDPS 能够有效增加用户就近访问数据的机会.

### (3) 成本节约率

成本节约率(cost saving rate, CSR)表示在所有作业请求中边缘节点完成的作业总费用与中心节点完成的作业总费用之比, 如式(23)所示.

$$CSR = \frac{\sum_{j=1}^n \sum_{i=1}^m T_{f_i}^{e_j \rightarrow u} \cdot SZ_{f_i} \cdot X_{f_i, e_j}}{\sum_{i=1}^m T_{f_i}^{c \rightarrow u} \cdot SZ_{f_i} \cdot \bar{X}_{f_i, e_j}}, \quad (23)$$

式中,  $T_{f_i}^{e_j \rightarrow u}$  表示从边缘节点  $e_j$  加载文件  $f_i$  到终端用户的单位费率,  $T_{f_i}^{c \rightarrow u}$  表示从中心节点加载文件  $f_i$  到终端用户的单位费率,  $T_{f_i}^{e_j \rightarrow u}$  通常是一个比  $T_{f_i}^{c \rightarrow u}$  小得多的数, 即  $0 < T_{f_i}^{e_j \rightarrow u} \leq T_{f_i}^{c \rightarrow u}$ . 一般而言, 随着文件访问次数的增加, 本地文件访问比例越大, 成本节约率指标越能够显著放大数据布局策略的成本效益. 为方便成本计算, 假定每次从云数据中心获取数据将产生一定的费用, 而从边缘节点获取数据则会节省此开销. 本文将 TDPS 与其他基准算法在不同并发请求规模下的成本节约率进行比较, 以衡量不同副本策略带来的经济效能, 如图 5 所示.

在图 5 中, 当并发请求规模较小时, TDPS 与其他基准算法的成本节约率差距不大, 各策略的成本节约率随着并发请求数的增加而增加; 当并发请求数达到 500 时, 与 HRS、D-ReP 和 RPME 3 个基准算法相比, TDPS 分别能够节约 2.68%、5.57% 和 8.43% 的成本开销; 之后, 随着并发请求规模继续增大, 不同算法的成本节约率虽时有波动, 但变化幅度始终处于一个较小范围, 总体维持动态平衡. 总体而言, TDPS 能够更显著地提高成本节约率, 原因在于: 一方面, TDPS 通过分布式副本推荐引擎为各边缘节点推荐个性化副本序列, 控制了局部副本规模; 另一方面, TDPS 基于副本放置规则学习模型优化面向全网的副本放置决策, 控制了全局副本规模. 因此, TDPS 采用“推荐-学习”的两级副本管理机制在成本控制方面的表现更优.

## 5 结论

本文提出一种基于“推荐-学习”的两阶段数据布局策略 TDPS, 解决云边环境下面向数据密集型应用如何改善用户 QoS 满意度并降低云边协同集群的管理和维护成本等核心问题. 采用基于移动预测和反馈优化的副本推荐引擎挖掘局部热点副本, 通过基于 A3C 强化学习的副本放置规则学习模型有机整合边缘

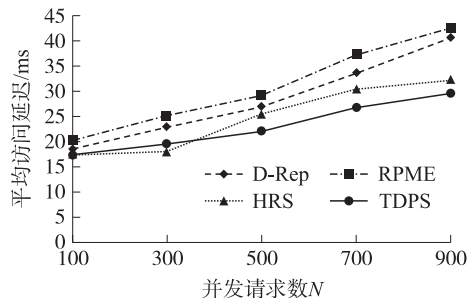


图 4 不同副本部署策略的平均访问延迟比较

Fig. 4 The comparison of average access latency for various replica deployment strategies

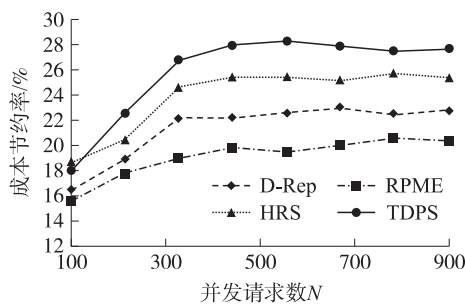


图 5 不同副本部署策略的成本节约率比较

Fig. 5 The comparison of cost saving rate for various replica deployment strategies

推荐信息,优化全网数据布局. 将 TDPS 与其他基准算法对比,实验结果表明,所提出策略在不同测试指标下都明显优于其他算法,验证了所提出方法的有效性. 下一步工作将考虑如何根据大规模分布式节点异构性调整数据布局策略,并在真实场景下对研究问题进行实验验证.

### [ 参考文献 ]

- [ 1 ] WANG T,ZHAO D,CAI S B,et al. Bidirectional prediction-based underwater data collection protocol for end-edge-cloud orchestrated system[J]. IEEE transactions on industrial informatics,2019,16(7):4791-4799.
- [ 2 ] 施巍松,张星洲,王一帆,等. 边缘计算:现状与展望[J]. 计算机研究与发展,2019,56(1):69-89.
- [ 3 ] LI A T,CHEN Y,YAN Z,et al. A survey on integrity auditing for data storage in the cloud:from single copy to multiple replicas[J]. IEEE transactions on big data,2022,8(5):1428-1442.
- [ 4 ] CHIANG M L,HSIEH H C,CHANG T Y,et al. An adaptive replica configuration mechanism based on predictive file popularity and queue balance in mobile edge computing environment[J]. Soft computing,2023,27(1):107-129.
- [ 5 ] AGIWAL M,ROY A,SAXENA N. Next generation 5G wireless networks:a comprehensive survey[J]. IEEE communications surveys & tutorials,2016,18(3):1617-1655.
- [ 6 ] ASIM M,WANG Y,WANG K,et al. A review on computational intelligence techniques in cloud and edge computing[J]. IEEE transactions on emerging topics in computational intelligence,2020,4(6):742-763.
- [ 7 ] 徐光伟,史春红,冯向阳,等. 基于多级网络编码的多副本云数据存储[J]. 计算机研究与发展,2021,58(2):293-304.
- [ 8 ] HE S W,REN J,WANG J H,et al. Cloud-edge coordinated processing:low-latency multicasting transmission[J]. IEEE journal on selected areas in communications,2019,37(5):1144-1158.
- [ 9 ] AFONSO N,BRAVO M,RODRIGUES L. Combining high throughput and low migration latency for consistent data storage on the edge[C]//Proceedings of the 29th Int Conf on Computer Communications and Networks. Piscataway,NJ:IEEE,2020.
- [ 10 ] LIU J W,SHEN H Y,NARMAN H S. Popularity-aware multi-failure resilient and cost-effective replication for high data durability in cloud storage[J]. IEEE transactions on parallel and distributed systems,2019,30(10):2355-2369.
- [ 11 ] ARAL A,OVATMAN T. A decentralized replica placement algorithm for edge computing[J]. IEEE transactions on network and service management,2018,15(2):516-529.
- [ 12 ] LI C L,WANG Y P,TANG H L,et al. Dynamic multi-objective optimized replica placement and migration strategies for SaaS applications in edge cloud[J]. Future generation computer systems,2019,100:921-937.
- [ 13 ] JIN J H,LI Y H,LUO J Z. Cooperative storage by exploiting graph-based data placement algorithm for edge computing environment[J]. Concurrency and computation:practice and experience,2018,30(20):1049-1064.
- [ 14 ] VALES R,MOURA J,MARINHEIRO R. Energy-aware and adaptive fog storage mechanism with data replication ruled by spatio-temporal content popularity[J]. Journal of network and computer applications,2019,135:84-96.
- [ 15 ] CHANG W C,WANG P C. Adaptive replication for mobile edge computing[J]. IEEE journal on selected areas in communications,2018,36(11):2422-2432.
- [ 16 ] TERANISHI Y,KIMATA T,YAMANAKA H,et al. Dynamic data flow processing in edge computing environments[C]//Proceedings of the 41st International Computer Software and Applications Conference. Piscataway,NJ:IEEE,2017:935-944.
- [ 17 ] XING J R,DAI H J,YU Z L. A distributed multi-level model with dynamic replacement for the storage of smart edge computing[J]. Journal of systems architecture,2018,83:1-11.
- [ 18 ] ARAL A,OVATMAN T. A decentralized replica placement algorithm for edge computing[J]. IEEE transactions on network and service management,2018,15(2):516-529.
- [ 19 ] WANG J D,ZHAO L,LIU J J,et al. Smart resource allocation for mobile edge computing:a deep reinforcement learning approach[J]. IEEE transactions on emerging topics in computing,2021,9(3):1529-1541.
- [ 20 ] DAI P L,HANG Z H,LIU K,et al. Multi-armed bandit learning for computation-intensive services in mec-empowered vehicular networks[J]. IEEE transactions on vehicular technology,2020,69(7):7821-7834.
- [ 21 ] TEREFE M B,LEE H,HEO N,et al. Energy-efficient multisite offloading policy using Markov decision process for mobile cloud computing[J]. Pervasive and mobile computing,2016,27:75-89.
- [ 22 ] WU H,LUO Y L,LI C L. Optimization of heat-based cache replacement in edge computing system[J]. Journal of supercomputing,2021,77(3):2268-2301.
- [ 23 ] SONMEZ C,OZGOVDE A,ERSOY C. EdgeCloudSim:an environment for performance evaluation of edge computing systems

- [C]//Proceedings of the 2nd Int Conf on Fog and Mobile Edge Computing. Piscataway, NJ:IEEE,2017:39-44.
- [24] MEDINA A,LAKHINA A,MATTA I,et al. BRITE:An approach to universal topology generation[C]//Proceedings of the 9th Int Symp on Modeling, Analysis and Simulation of Computer and Telecommunication Systems. Piscataway, NJ:IEEE,2001:346-353.
- [25] TANG Y Y,WANG H,GUO K H,et al. A new replica placement mechanism for mobile media streaming in edge computing[J]. Concurrency and computation:practice and experience,2019,33(7):e5361.
- [26] MANSOURI N,JAVIDI M. A hybrid data replication strategy with fuzzy-based deletion for heterogeneous cloud data centers[J]. Journal of supercomputing,2018,74(10):5349-5372.

[责任编辑:黄 敏]