

协作缓存及数学建模在 CCN 中的应用

李 伟,潘沛生

(南京邮电大学 通信与信息工程学院,江苏 南京 210003)

摘 要:为了解决当前网络暴露出来的诸多问题,对未来网络架构的研究已经形成一个新的领域。而内容中心网络则是这个领域中非常重要的研究对象。在内容中心网络中,每个节点都具备缓存数据的能力,因此缓存策略对内容中心网络的性能具有决定性的影响。通过对现有缓存策略的调研发现,网络中的各节点之间缺少协作机制,相邻节点不存在通信,也就不能较为智能地选择已存储请求内容的最近节点。基于这种现状,对基于内容流行度的自治系统协作缓存在网络中的应用进行了研究,将网络划分为多个自治系统,确定各自的控制节点,协调所属系统中各节点之间的通信,以求得到更好的网络性能。对任意的网络拓扑结构进行数学建模,求解并比较现有策略和基于自治系统的协作缓存策略的三种指标,包括平均跳数、命中率和时延。仿真结果表明,该自治系统协作缓存策略的应用使得网络性能更佳。

关键词:内容中心网络;自治系统;协作缓存;控制节点;任意拓扑;数学建模

中图分类号:TP31

文献标识码:A

文章编号:1673-629X(2018)07-0038-05

doi:10.3969/j.issn.1673-629X.2018.07.009

Application of Cooperative Caching and Mathematical Modeling in CCN

LI Wei, PAN Pei-sheng

(School of Communication and Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing 210003, China)

Abstract: In order to solve the problems exposed by the current network, the study of architecture in the future network has formed a new area for research. The content-center network (CCN) is becoming a research hot spot in this area. Each node in CCN has the ability to cache data, so the caching strategy has a decisive impact to the performance of network. The survey shows that the nodes in CCN lack co-operation mechanism in the existing caching strategies, and it is impossible to choose the nearest node storing required contents intelligently. With the analysis of these problems, we propose an autonomous system collaboration caching strategy based on the content popularity (P-ASS). The network is divided into multiple autonomous systems in P-ASS, and the leading node is determined to coordinate communications among nodes to get better performance. In order to make the conclusion more universal, the simulation is performed on any topology. Average hops, hit ratio and delay time are calculated and compared between some existing strategies and P-ASS. The result of simulation shows that P-ASS will improve the cache efficiency.

Key words: CCN; autonomous system; collaborative cache; control nodes; random topology; mathematical modeling

0 引 言

随着信息量的暴涨,当前网络架构已经不能满足用户的需求,对未来网络架构的研究开始变得愈发重要。近年来,学术界提出的 CCN 网络,成为未来网络架构领域中值得关注的热点^[1]。目前,CCN 有两种最基础的缓存策略,分别为处处缓存策略 (leave cache everywhere, LCE)^[2] 和向下拷贝策略 (leave copy down, LCD)^[3]。这两种策略简单易行,但资源的利用

率较低^[4]。

为了解决节点之间完全独立的现状,对如何把自治系统的协作策略应用到 CCN 网络中进行了研究。能够实现自治系统内部各节点之间互相通信,使得节点收到请求之后,能够更加快速智能地选择存储该请求内容的节点,而不是直接使用最短路径优先策略来决定请求转发路线,从而以较少的跳数和较小的延时得到请求内容。对自治系统协作缓存策略 (P-ASS)

收稿日期:2017-07-16

修回日期:2017-11-22

网络出版时间:2018-02-24

基金项目:江苏省高校自然科学基金基础研究计划重大项目 (08KJD510001)

作者简介:李 伟 (1992-),女,硕士,研究方向为新一代网络缓存;潘沛生,博士,教授,研究方向为新一代无线通信。

网络出版地址: <http://www.cnki.net/kcms/detail/61.1450.TP.20180224.1510.018.html>

进行定性分析。在实现自治系统集中控制之前,通过将网络划分为自治系统来实现显式协作,减少缓存的冗余,提高缓存利用率和缓存命中率。在此策略下,同一 AS 中的节点相互合作,实现了 CCN 的缓存性能的增益。

为了使 P-ASS 更具普遍性,对 P-ASS 进行定量计算时,针对任意网络拓扑建立数学模型,而不是仅针对二叉树等几种较为普遍的拓扑图,使得仿真结果更具普遍性。最后计算并比较 LCE、LCD 及 P-ASS 的缓存命中率、平均跳数和延时时间,以验证 P-ASS 的性能。

1 自治系统协作缓存策略

1.1 基本思想

缓存策略对 CCN 的性能有决定性的影响。在传统的 CCN 中,数据包沿返回路径被缓存,并且只有传输路径上的缓存可以用于服务响应,缺乏节点之间必要的协作机制,使得系统缓存效率很低。此外,从相邻节点获取内容的成本比从源服务器获取它的成本便宜得多。而提出的协作缓存策略解决了节点之间缺乏有效协作的问题,使得缓存得到了更有效的利用,以达到提高系统缓存效率的目的。

1.2 P-ASS 详述

1.2.1 AS 的划分

在 P-ASS 中,缓存系统分为几个由控制节点集中控制的 AS,如图 1 所示。自治系统内仍然使用 OSPF 协议^[5],利用最短路径优先算法决定请求路径。

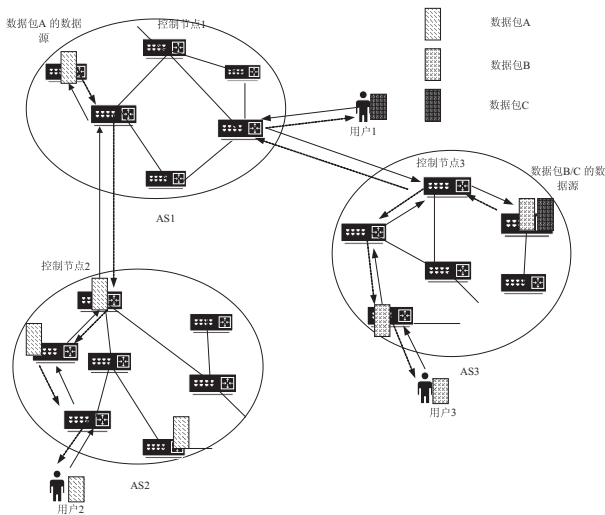


图 1 任意网络拓扑

1.2.2 控制节点的选择

控制节点的选择决定网络的性能,因此如何选择控制节点变化变得尤为重要。下面介绍一种使用基于节点的中间性和缓存替换率^[6]来选择控制节点的方法。 万方数据

CCN 网络的拓扑结构可以表示为无向图 $G = (V, E)$, 其中 V 是 CCN 节点的集合, E 是节点之间的边集。 $C(v)$ 是与节点 v ($v \in V$) 相连接的节点个数。一旦网络建立,就很容易获得 $C(v)$ 。 $Cnor(v)$ 是 $C(v)$ 的归一化,可以通过式 1 求得。

$$Cnor(v) = \frac{C(v)}{\max_{v \in V} C(v)}, Cnor(v) \in [0, 1] \quad (1)$$

关于节点 v ,被替换的内容 k_i 的大小由 $S(k_i)$ 表示,节点 v 的缓存大小由 $Ca(v)$ 表示。 m 是单位时间内节点 v 的缓存内容替换次数。 $Re(v)$ 是节点 v 的缓存替换率,见式 2。

$$Re(v) = \frac{\sum_{i=1}^m S(k_i)}{Ca(v)}, Ca(v) \neq 0 \quad (2)$$

$Re(v)$ 归一化后的值表示为 $Renor(v)$, 见式 3:

$$Renor(v) = \frac{Re(v)}{\max_{v \in V} (Re(v))}, Renor(v) \in [0, 1] \quad (3)$$

$M(v)$ 表示网络中每个节点作为控制节点的适应度情况,由式 4 求得。

$$M(v) = \frac{Cnor(v)}{Renor(v)}, Renor(v) \neq 0 \quad (4)$$

1.2.3 流行度计算及对应策略

在 P-ASS 中,AS 的控制节点需要具有计数内容的流行度^[7]的能力。使用式 5 和式 6 计算每个内容的流行度。

$$Popu(i) = \frac{R_i}{\Delta t}, i \in I \quad (5)$$

$$Popularity(i) = \frac{Popu(i)}{\max_{i \in I} (Popu(i))}, Popularity(i) \in [0, 1] \quad (6)$$

其中, i 表示请求内容的名称; R_i 表示用户在时间段 t 内请求内容 i 的次数^[8]; I ($i \in I$) 表示时间间隔 t 期间的所有内容的集合。

实际上, $Popu(i)$ 是流行度,但是式 6 对 $Popu(i)$ 进行归一化,得出 $Popularity(i)$ 。 $Popularity(i)$ 在以后更方便使用。

文中按照内容流行度参数将内容分为三类:高流行度内容、中等流行度内容和低流行度内容。不同类型的内容具有由控制节点确定的不同的缓存策略。如果 $Popularity(i) > 0.5$,则内容 i 是高流行度内容。它需要冗余的缓存来提高缓存命中率,因为许多用户会请求它。图 1 的内容 A 就是一个示例。如果 $0.2 < Popularity(i) < 0.5$,这意味着内容 i 是中等流行度内容。中等流行度内容是减少冗余的主要部分。在 P-ASS 中,这些内容在同一 AS 中仅会被缓存一次,以减

少内容缓存冗余。图 1 中的 B 是中等流行度内容。如果 $Popularity(i) < 0.2$, 则内容 i 被称为低流行度内容。因为低流行度内容被请求的次数太少, 容易被替换^[9], 如图 1 中的 C, 所以它们通常不在 AS 中缓存。

1.3 AS 中的节点详述

1.3.1 AS 中各节点的结构

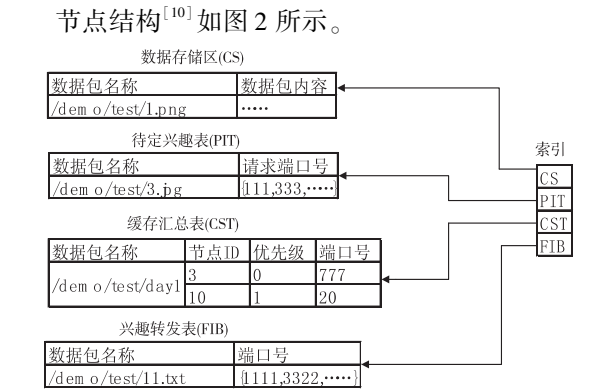


图 2 CCN 节点结构

在 AS 中有两种类型的节点, 分别是控制节点和公共节点。AS 中只有一个控制节点, 而其他都是公共节点。控制节点的主要作用是控制整合该自治系统所有节点的存储内容, 公共节点则定期向控制节点更新自己本地的信息。

下面是两种类型节点的具体作用:

(1) 控制节点: 除了传统的三个表^[11] (CS, PIT, FIB) 之外, 特别地, 每个控制节点维护自己的缓存汇总表 (CST)。它记录所属 AS 中各节点缓存的内容信息。每个节点周期性地向控制节点报告其本地缓存信息。控制节点周期性地将自己的 CST 通告给公共节点。同时, 控制节点计算自己已经接收的每个内容的流行度以确定不同内容的缓存策略。

(2) 公共节点: 除了控制节点之外, AS 中的其他节点是公共节点。公共节点保持四个表: CS, PIT, FIB 和 CST, 用于记录内容的名称及其获得的位置。然而, LCST 由控制节点发出。当新内容已被缓存时, 公共节点向控制节点报告 CS 的信息。

1.3.2 通信包类型及作用

P-ASS 是基于节点之间的相互通信提出的, 则节点之间的通信内容也会有所不同。在该策略中定义了六种类型的通信包, 供不同的消息类型使用。分别是: 兴趣包、数据包、汇报包、更新包、探测包和确认包。

(1) 兴趣包: 兴趣包由用户发送, 用于请求想要的网络资源等。

(2) 数据包: 数据包是当用户请求在网络中命中时, 返回给用户的数据信息。

(3) 汇报包: 公共节点在更新存储的内容时, 会发送汇报包给控制节点, 通知控制节点更新汇总表记录。

如果一定时间段内没有存储内容的更新, 节点会定期发送汇报包, 告诉控制节点, 该节点还存在。

(4) 更新包: 控制节点周期性地将自己 CST 中的汇总信息同步给所属自治系统中的各个公共节点。如果请求在控制节点命中, 则控制节点立刻将请求转发给相应的公共节点。

(5) 探测包: 当公共节点不能在自己的缓存中命中请求内容, 则会将请求转发给控制节点, 请求自己想要的的内容。如果一定时间内, 该节点没有收到控制节点的确认包或者数据包, 则该节点会向控制节点发送探测包, 以确认控制节点是否已收到该节点的请求。

(6) 确认包: 当控制节点收到请求包时, 控制节点在自己的 CS, PIT, FIB 和 CST 中查看该内容。如果控制节点从请求该内容的节点处收到探测包, 则会发送确认包, 告诉该节点请求有没有命中。

1.3.3 基本通信过程

情况 1: 如果控制节点收到用户请求, 首先计算该内容的流行度以确定哪些节点适于缓存该内容, 以提高缓存利用率并增加缓存命中率。之后, 作为传统的 CCN, 它应该依次查找 CS、PIT、CST、FIB。如果缓存未命中, 则控制节点将丢弃兴趣包并发送确认包以通知请求该内容的节点自己处理。

情况 2: 如果接收机是公共节点, 则应当搜索其 CS、PIT、LCST 作为传统 CCN 来检查其是否具有内容。如果没有匹配, 则公共节点将向控制节点转发兴趣包。然后控制节点将像情况 1 一样处理。公共节点在将兴趣包转发到控制节点之后等待数据包, 如果等待超时, 则公共节点将发送消息以确认内容是否命中。如果公共节点接收到控制节点的回复, 并且被告知内容未命中, 则公共节点将如传统 CCN 那样检查 FIB。

此外, 如果兴趣包能够匹配到 CST 中的内容, 则说明内容已被 AS 中的节点缓存, 则会将兴趣包转发到该节点以获得内容。

2 对任意网络拓扑建模

在提出新的缓存策略之后, 本节将针对任意网络拓扑结构进行数学建模^[12]。将网络的数据传输通过概率论原理, 用数学表达式表述。考虑 CCN 网络的请求聚合能力, 建立 CCN 在一般网络拓扑结构下, 基于迭代方法的 MMAT 传输模型。然后将所列数学表达式进行迭代, 直到未知量落入预先设定的误差范围终止。最后得到内容的平均未命中率和时延。在该模型上, 先后使用 LCE、LCD 和 P-ASS 三种策略进行仿真。使用相同的计算模型, 保证仿真数据更可靠, 结论更客观。

符号及意义见表 1。

表1 符号及意义

符号	意义
$B_{i,j}$	相邻节点 v_i 和 v_j 之间的带宽
$\lambda_{k,v}$	内容 c_k 在节点 v 的请求率
$m_{k,v}$	内容 c_k 在节点 v 的未命中率
$f_{k,v}$	内容 c_k 的请求在节点 v 被转发率
$P_{k,v}$	节点 v 获取内容 c_k 的路径集
$P_{k,v}(l)$	路径集 $P_{k,v}$ 的第 l 跳节点集
$\text{hop}_{k,v}(l)$	路径集 $\text{hop}_{k,v}$ 的第 l 跳节点集
$t_{i,j}$	v_i 和 v_j 获取内容 c_k 的往返时延
$\text{VT}_{k,v}$	内容 c_k 在节点 v 上获得的虚拟往返时延

两相邻节点 v_i 和 v_j 之间获取的内容 k 的往返时延是:

$$t_{v_i,v_j} = \begin{cases} 2t_p + t_q + t_c, e_{i,j} = 1 \\ 0, e_{i,j} = 0 \end{cases} \quad (7)$$

其中, t_p 为相邻节点的传播时延; t_q 为请求传输时延; t_c 为内容传输时延。则节点 v 按照 $P_{k,v}$ 在第 j 跳节点获取内容 c_k 的往返时延表示为:

$$T_{k,v,j} = t_{v,P_{k,v}(2)} + t_{P_{k,v}(2),P_{k,v}(3)} + \cdots + t_{P_{k,v}(j-1),P_{k,v}(j)} \quad (8)$$

此时,便可得到 $\text{VT}_{k,v}$ 为 $T_{k,v,j}$ 命中概率的加权。

下面对所提建模方案进行详细描述。设 $G(V,E)$ 为一个 CCN 网络,将网络中的每个节点看作由 CS、PIT、FIB 这三个过滤器组成。则具体建模步骤如下:

第一步:节点 v 收到对内容 k 的请求率为:

$$R_{k,v} = \lambda_{k,v} + \sum_{v' \in \text{hop}_{k,v}(2)} f_{k,v'} \quad (9)$$

第二步:计算请求在经过 CS 后的输出流。根据文献[13]可得请求内容在节点 v 处的概率为:

$$\vec{q}_v = \text{SCA}(\text{policy}, \vec{r}_v, |v|) \quad (10)$$

其中, $\vec{q}_v = \{q_{1,v}, q_{2,v}, \cdots, q_{N,v}\}$, $q_{k,v}$ 表示内容 k 的请求在节点 v 处命中的概率。则

$$\text{VT}_{k,v} = \sum_{j=1} d_{k,v,j} \left(\sum_{v' \in P_{k,v}(j)} q_{k,v'} \right) \prod_{l=1}^{j-1} \left(1 - \sum_{v' \in P_{k,v}(l)} q_{k,v'} \right) \quad (11)$$

假设请求流服从泊松分布,根据文献[14]可得内容 k 的请求在节点 v 处未命中的概率为:

$$m_{k,v} = R_{k,v} (1 - q_{k,v}) \quad (12)$$

第三步:请求通过 CS 到达 PIT 之后,PIT 过滤被转发之后还未收到回复的请求。则节点 v 处请求的聚合概率取决于 PIT 中记录的生存时间 T 和 $\text{VT}_{k,v}$ 。当数据在 CS 中的缓存时间远远小于 $\text{VT}_{k,v}$ 时,到达 PIT 的请求也满足泊松分布,因此节点 v 请求内容 k 的聚合概率为: $(1 - e^{-m_{k,v} \Delta_{k,v}})$,其中 $\Delta_{k,v} = \min(T, \text{VT}_{k,v})$ 。

第四步:将式7~12联立,所有节点的CS和 $f_{k,v}$ 全部清零,然后迭代,直到两次相邻迭代的平均跳数、命中率和平均往返时延都小于所设定的阈值。

3 仿 真

为了分析 P-ASS 的性能,使用 ccnSim 平台^[15]进行仿真。仿真网络中设有 400 个节点,两节点间带宽为 20 Mbps,传播时延 $t_p = 5 \times 10^{-4} \text{ s}$,内容大小为 1 kb。本次性能评估的主要内容是,不同的网络缓存大小对各种策略性能的影响。与 CCN 中的两种传统缓存策略(LCE、LCD)进行比较,结果如图3~5所示。

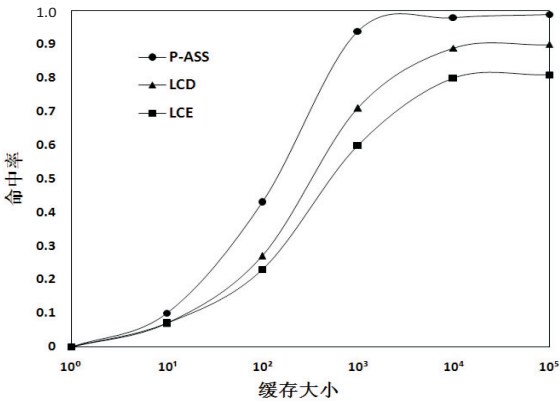


图3 命中率随缓存大小的变化

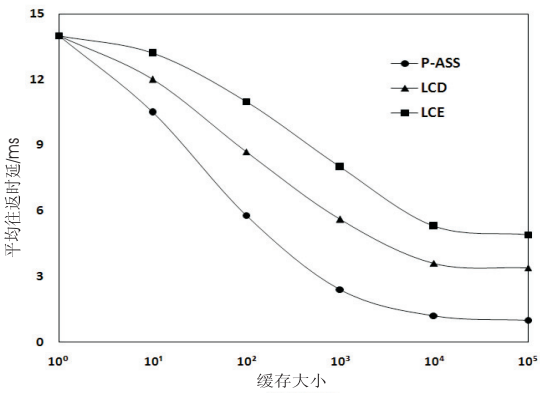


图4 平均往返时延随缓存大小的变化

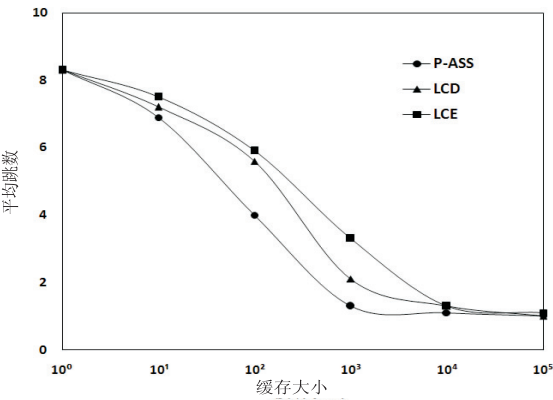


图5 平均跳数随缓存大小的变化

通过数据对比可以看出,与基本的 LCE、LCD 策略相比,随着缓存大小的增长,P-ASS 在平均跳数、命中率和平均时延三方面的变化更具备优越性。在缓存大小趋于无限大时,三种策略性能参数都趋于稳定,但

P-ASS 性能最佳。

4 结束语

文中对 CCN 的两种基本缓存策略进行调研,得出几种策略的不足之处,比如缓存冗余度高,只能沿单一路径请求内容,且节点之间相互独立,缺少必要的协作通信,等等。在此基础上,提出一种基于自治系统协作缓存的策略,使得相邻节点之间可以实现相互通信,比较智能地就近选择合适的节点,获取用户请求的内容,实现以较少的跳数实现内容的命中。三种策略在面向任意拓扑的同一传输模型上进行仿真实验,结果表明,与现有基本策略相比,P-ASS 可以得到更好的网络性能参数,优化网络性能。未来网络架构还有很多方面值得研究,例如可以根据当前网络环境及内容存储的位置,更加智能地动态选择控制节点,以实现更好的性能增益,收获更好的用户体验。

参考文献:

- [1] 吴建平,林 嵩,徐 恪,等.可演进的新一代互联网体系结构研究进展[J].计算机学报,2012,35(6):1094-1108.
- [2] 李春洪,冯国富,顾铁成,等.一种无“热点”的覆盖网协同缓存策略[J].软件学报,2008,19(3):744-754.
- [3] 李文中,陈道蓄,陆桑璐.分布式缓存系统中一种优化缓存部署的图算法[J].软件学报,2010,21(7):1524-1535.
- [4] PARISIS G, TROSSEN D, SYRIVELIS D. Implementation and evaluation of an information-centric network[C]//IFIP networking conference. Brooklyn, NY, USA: IEEE, 2013: 1-9.
- [5] 潘 楠,王 勇,陶晓玲.基于 OSPF 协议的网络拓扑发现算法[J].计算机工程与设计,2011,32(5):1550-1553.
- [6] TARNOI S, SUKSOMBOON K, KUMWILAI SAK W, et al.

Performance of probabilistic caching and cache replacement policies for content-centric networks[C]//IEEE conference on local computer networks. Edmonton, Canada: IEEE, 2014: 99-106.

- [7] XU Yuemei, MA Shuai, LI Yang, et al. P-CLS: a popularity-driven caching location and searching scheme in content centric networking [C]//IEEE 34th international performance computing and communications conference. Nanjing, China: IEEE, 2015: 1-8.
- [8] GARETTO M, LEONARDI E, TRAVERSO S. Efficient analysis of caching strategies under dynamic content popularity [C]//IEEE conference on computer communications. Hong Kong: IEEE, 2015: 2263-2271.
- [9] 王 楠. 基于 CCN 的路由优化策略研究[D]. 北京: 北京交通大学, 2014.
- [10] WANG Jia. A survey of web caching schemes for the Internet [J]. ACM SIGCOMM Computer Communication Review, 1999, 29(5): 36-46.
- [11] 黄慧群. 内容中心网络的查表技术研究[D]. 郑州: 解放军信息工程大学, 2014.
- [12] 王国卿. 内容中心网络建模与内容放置问题研究[D]. 北京: 北京邮电大学, 2014.
- [13] DAN A, TOWSLEY D. An approximate analysis of the LRU and FIFO buffer replacement schemes [J]. ACM SIGMETRICS Performance Evaluation Review, 1990, 18(1): 143-152.
- [14] ROSENSWEIG E J, KUROSE J, TOWSLEY D. Approximate models for general cache networks [C]//INFOCOM. [s. l.]: IEEE, 2010: 1-9.
- [15] CHIOCCHETTI R, ROSSI D, ROSSINI G. ccnSim: an highly scalable CCN simulator [C]//IEEE international conference on communications. Budapest, Hungary: IEEE, 2013: 2309-2314.

(上接第 24 页)

- [7] MAZUMDER R, HASTIE T, TIBSHIRANI R. Spectral regularization algorithms for learning large incomplete matrices [J]. Journal of Machine Learning Research, 2010, 11: 2287-2322.
- [8] LIU Z. Collaborative filtering recommendation algorithm based on user interests [J]. International Journal of u- and e-Service, Science and Technology, 2015, 8(4): 311-320.
- [9] 康钟荣. 基于项目分类与预测填充的协同过滤推荐算法研究[D]. 北京: 北京化工大学, 2013.
- [10] DAI Jisheng, HU Nan, XU Weichao, et al. Sparse Bayesian learning for DOA estimation with mutual coupling [J]. Sensors, 2015, 15(10): 26267-26280.
- [11] SARWAR B, KARYPIS G, KONSTAN J, et al. Application of dimensionality reduction in recommender systems [C]//

ACM Webkdd workshop. [s. l.]: ACM, 2000: 1753-1765.

- [12] 孙光福, 吴 乐, 刘 淇, 等. 基于时序行为的协同过滤推荐算法 [J]. 软件学报, 2013, 24(11): 2721-2733.
- [13] CHOI K, SUH Y. A new similarity function for selecting neighbors for each target item in collaborative filtering [J]. Knowledge-Based Systems, 2013, 37: 146-153.
- [14] 张光卫, 李德毅, 李 鹏, 等. 基于云模型的协同过滤推荐算法 [J]. 软件学报, 2007, 18(10): 2403-2411.
- [15] LUO Heng, NIU Changyong, SHEN Ruimin, et al. A collaborative filtering framework based on both local user similarity and global user similarity [J]. Machine Learning, 2008, 72(3): 231-245.
- [16] KALELI C. An entropy-based neighbor selection approach for collaborative filtering [J]. Knowledge-Based Systems, 2014, 56: 273-280.