doi:10.11959/j.issn.1000-436x.2015276

基于内容流行度差异性的 CDN-P2P 融合分发网络缓存替换机制研究

聂华，张敏，郭敬荣，阳小龙
(北京科技大学 计算机与通信工程学院，北京 100083)

摘 要：现有的 CDN-P2P 缓存替换机制没有关注内容文件中各片段的个体流行度差异性，而无法提高预缓存内容片段的访问命中率。鉴于此，提出了基于流行度差异性的缓存替换机制 Diff-Attribute。同时考虑了内容文件的整体流行度和文件中各个片段的个体流行度。此外，基于分布熵，定义了一种内容流行度均衡性度量方法：若流行度均衡，就提前缓存各文件的前缀片段；否则提前缓存热门文件或其中最热门的内容片段。仿真结果表明：在缓存命中率和字节命中率方面，Diff-Attribute 机制分别高出 LFU、LRU 等传统机制约 6%和 8%；在访问延迟启动率和传输成本消耗率方面，Diff-Attribute 机制则降低了约 13%和 7%。
关键词：内容分发网络；内容缓存；P2P 网络；流行度；分布熵
中图分类号：TP393 **文献标识码**：A

Content popularity difference-aware cache eviction scheme for CDN-P2P hybrid networks

NIE Hua, ZHANG Min, GUO Jing-rong, YANG Xiao-long
(School of Computer and Communications Engineering, Beijing University of Science and Technology, Beijing 100083, China)

Abstract: In CDN-P2P hybrid network, it is important for the cache eviction schemes to improve the delivery efficiency of content. However, most of them only consider the holistic popularity of content file, and neglect the difference between the individual popularities of segments within a content file. Hence, it was difficult to improve the hit rate of pre-cached content segments, and to reduce the user access delay. Hence based on the difference between the attributes of content popularity, a new cache eviction scheme (i.e., Diff-Attribute) was proposed. Bewildered the holistic popularity of a content file, it also considered the individual popularity of its segment. More importantly, based on the concept of entropy, A method to measure the popularity difference between content files or segments was put forward. If the popularities of the segments within a content file are equalizing, its prefix segment would be pre-cached. Otherwise, the requested segments or files directly based on its popularity would be cached. Compared with traditional schemes (e.g., LFU, LRU, MRU,FIFO), the simulation results show that Diff-Attribute can improve the cache hit rate and the byte hit rate by at least 6%, 8% respectively, and can reduce the access startup delay rate and the transmission cost rate by at least 13%, 7% respectively.
Key words: content distribution network; cache eviction; peer-to-peer network (P2P); content popularity; entropy

1 引言

CDN-P2P 融合分发网络结合了 CDN 和 P2P 这 2 种技术的优点，是当前 IP 网络提供内容共享服务的主流服务网络形态[1,2]。CDN-P2P 网络的核心工作模式如下：首先将流媒体内容以 CDN 集中控制方式分发到各用户自治域的副本服务器，然后再由副本服务器将内容以 P2P 分布式推送到各用户自治域。在内容分发过程，副本服务器中缓存内容替换是否优化，则严重影响到用户访问命中率和用户访问响应时间等关键性能和用户体验质量指标[3]。因此缓存替换是 CDN-P2P 内容分发网络的

收稿日期：2015-10-10
基金项目：国家自然科学基金资助项目 (61172048, 61100184)
Foundation Item: The National Natural Science Foundation of China (61172048, 61100184)

研究重点之一。

针对缓存替换问题，目前人们提出了许多解决方案，其中典型的有 LRU(least recently used)算法、LFU(least frequently used)算法和 MRU(most recently used)算法等[4~6]。LRU 和 LFU 算法分别根据访问近期性和访问频率进行缓存替换。虽然它们的实现都很简单，但是它们都各自存在不同的问题，其中 LRU 存在长环模式问题，即内容的重用模式长度可能大于缓存空间大小，使刚被替换出缓存空间又被请求访问；而 LFU 未考虑时间局部性，过去访问频率高的内容即使不再被访问也不能被替换，造成缓存空间浪费。MRU 与 LRU 相反，它将刚刚被访问的数据替换掉，因此 MRU 比较适合于顺序访问或循环访问。这些缓存替换机制只考虑了访问时间、频率等局部性因素，而没有考虑访问内容的流行度。通常对于一个内容文件(如视频)，用户会略过其片头和片尾，而重点关注其中剧情跌宕起伏的片段。然而上述机制均是对一个内容文件整体进行缓存替换操作，因此不仅替换操作开销大，缓存资源有效利用率低，而且也无法降低用户内容访问延迟。

为此，文献[7]认为每个内容文件可以根据其内容特征划分为不同的组成片段，而提出了一种以内容片段为操作对象的缓存替换机制。它能根据每个内容片段的流行度高低进行缓存替换，但是它没有考虑如何优化替换，尤其是对初始访问内容的提前缓存，因此它的初始访问延迟较大。而文献[8]提出了一种基于重用时间的缓存替换机制，它将最久没被访问的内容片段替换掉。然而该机制只考虑最久没被访问时间，而没有考虑其中各片段的个体流行度差异性。尤其在内容片段被访问次数和片段所在内容文件的访问流行度相同情形下，该机制无法区分哪些缓存片段该被替换。

实际上，大量研究结论表明：流媒体内容的访问流行度分布符合 zipf 定律，约 20%的流媒体内容被超过 80%的用户请求访问，且一个流媒体文件的各个内容片段的访问流行度也不相同[9~11]。然而，如何基于各内容片段访问流行度差异性或不均衡性，优化缓存片段替换，则是现有机制均没有考虑的。为此，本文将以内容文件或内容片段访问流行度为决策依据，提出基于内容访问属性差异性的缓存替换机制 Diff-Attribute。

2 基于内容访问属性差异性的缓存替换机制

Akamai 调查研究发现[12]：流媒体文件的访问流行度呈现非均衡性，即同一个文件中各内容片段的流行度各不相同。该均衡性可采用流行度分布熵来衡量，因为熵能够定量地描述流行度偏离于均匀分布的程度。当所有片段的流行度相差不大时，其熵值最大；反之，当流行度分布各不相同时，其熵值最小。若用户对某个流媒体内容文件从开头看到结尾，则表明该内容文件内各片段的流行度分布均衡。此时，若提前缓存该文件的前缀片段，则能够大大降低用户访问延迟。相反地，若用户仅观看该文件内的部分片段，则表明该内容文件内各片段的流行度分布不均衡。此时，系统只需缓存该文件中流行度较高的内容片段，以保证被缓存内容具有较高的流行度。因此，Diff-Attribute 机制的设计思路如下：设置适当的临界熵值 Φ，根据流行度分布情况对内容片段分类：当内容的流行度分布熵不小于 Φ 时，则该内容各片段流行度分布相对均衡，提前缓存它的前缀片段；当内容的流行度分布熵小于 Φ 时，则其流行度分布不均衡，而只缓存流行度较高的内容片段，保证缓存空间中的内容具有最高的流行度。

2.1 流行度分布熵的计算

假设 CDN-P2P 分发网络中内容文件集为 $D=\{C_k|1\leqslant k\leqslant K\}$。根据其内容语义和用户访问习惯等特征，一个内容文件 C_k 可分为 X 个片段，即 $C_k=\{S_{k,i}|1\leqslant i\leqslant X\}$。针对每一个内容片段 $S_{k,i}$，副本服务器维护了如表 1 所示的访问日志。

表 1 各内容片段相关的访问日志信息

访问参数	具体描述
$T^{(i)}_{\text{recent}}$	内容片段 $S_{k,i}$ 最近一次被访问的时间
$T^{(i)}_{\text{first}}$	内容片段 $S_{k,i}$ 第一次被访问的时间
$M^{(i)}$	副本服务器中内容片段 $S_{k,i}$ 被访问的总次数，即命中次数
N	副本服务器收到的对所有内容片段的请求次数

根据信息熵概念，内容文件 C_k 中各片段访问流行度的均衡性（即流行度分布熵）定义如下

$$H(C_k)=-\sum_{i=1}^{X}P(S_{k,i})\text{lb}^{P(S_{k,i})} \tag{1}$$

其中，$P(S_{k,i})$ 为内容片段 $S_{k,i}$ 的流行度，它常常受到各种因素的影响，最主要包括用户访问的时间局部性和用户 VCR 交互操作，而时间局部性

因素主要体现在内容访问频率、访问近期性和平均访问时间间隔等如表 1 所示的访问日志信息。综合这 2 大因素，内容片段 $S_{k,i}$ 的流行度的计算为

$$P(S_{k,i})=\frac{\dfrac{M^{(i)}}{N}\dfrac{1}{T^{(i)}-T^{(i)}_{\text{recent}}+1}\dfrac{B^{(i)}+\alpha}{F^{(i)}+\alpha}}{\dfrac{T^{(i)}_{\text{recent}}-T^{(i)}_{\text{first}}}{M^{(i)}}} \tag{2}$$

其中，$T^{(i)}$ 是内容片段 $S_{k,i}$ 被访问的当前时刻，$\dfrac{M^{(i)}}{N}$ 表示内容片段 $S_{k,i}$ 被访问的频率，$\dfrac{1}{T^{(i)}-T^{(i)}_{recent}+1}$ 表示内容片段 $S_{k,i}$ 被访问的近期性：若 $S_{k,i}$ 刚刚被访问，访问近期性较大；若很久未被访问，则访问近期性较小，从一定程度上避免了刚刚被缓存的内容又被替换出去。$\dfrac{T^{(i)}_{\text{recent}}-T^{(i)}_{\text{first}}}{M^{(i)}}$ 表示平均访问时间间隔：若该值较大，表示内容片段的访问频率下降；若该值较小，则表示目前内容片段仍有较高的访问频率。$\dfrac{B^{(i)}+\alpha}{F^{(i)}+\alpha}$ 是 VCR 交互操作因子，其中，$B^{(i)}$、$F^{(i)}$ 分别表示用户对该内容片段 $S_{k,i}$ 施加的向后回放操作和向前拖拽操作的次数，而 α 是一个接近零的常数，以避免 $B^{(i)}$、$F^{(i)}$ 为 0 时流行度公式失去意义。$B^{(i)}$ 越大，表示用户对 $S_{k,i}$ 越感兴趣；$F^{(i)}$ 越大，表示用户对 $S_{k,i}$ 不感兴趣。

同理对一个内容文件 C_k，它的整体流行度 $P(C_k)$ 为

$$P(C_k)=\frac{\dfrac{\sum_{i=1}^{X}M^{(i)}}{N}\dfrac{1}{T^{(i)}|_X-T^{(i)}_{\text{recent}}|_X+1}\dfrac{\sum_{i=1}^{X}B^{(i)}+\alpha}{\sum_{i=1}^{X}F^{(i)}+\alpha}}{\dfrac{T^{(i)}_{\text{recent}}|_X-T^{(i)}_{\text{first}}|_X}{\sum_{i=1}^{X}M^{(i)}}} \tag{3}$$

其中，$T^{(i)}|_X$ 表示用户正在访问文件 C_k 中任一片段的当前时间，即 $T^{(i)}|_X=\text{Max}\{T^{(i)}\},\forall i\in[1,X]$；而 $T^{(i)}_{\text{first}}|_X$ 则表示文件 C_k 中所有片段中最早被访问的时间，即 $T^{(i)}_{\text{first}}|_X=\text{Min}\{T^{(i)}_{\text{first}}\},\forall i\in[1,X]$。

为了简单判断内容文件 C_k 中各片段的访问流行度是否均衡，可为其设置熵临界值 Φ。若内容文件 C_k 的流行度分布熵 $H(C_k)$ 高于临界值 Φ，则其片段流行度分布 $P(S_{k,i})$ 趋于平衡；反之，其片段流行度分布严重不均衡。这里，Φ 有以下 3 种不同取值策略。

1) 最小取值策略

$$\Phi=\text{Min}\{H(C_k)\},\ \forall k\in[1,X] \tag{4}$$

2) 最大取值策略

$$\Phi=\text{Max}\{H(C_k)\},\ \forall k\in[1,X] \tag{5}$$

3) 内容文件集 D 流行度分布熵取值策略

假定副本服务器内容文件集 D 中各文件 C_k 的整体流行度为 $P(C_k),\forall j\in[1,K]$，则对该文件集 D 流行度分布熵定义如下

$$\Phi=-\sum_{k=1}^{K}P(C_k)P(C_k) \tag{6}$$

因此，Φ 值与该文件 C_k 的流行度分布熵 $H(C_k)$ 的相对大小决定了该文件的前缀片段是否应被缓存。若内容文件 C_k 内各片段的流行度分布均衡，则用户极大可能会对该文件从头看到尾。此时，若提前缓存该文件的前缀片段，则能够大大降低用户访问延迟。相反地，若文件内各片段的流行度分布不均衡，则用户仅观看该文件内的部分片段，尤其是其中流行度较高的片段。此时，系统只需缓存该文件中流行度较高的内容片段，以提高缓存资源利用率，降低缓存替换操作开销。

2.2 Diff-Attribute 缓存替换机制

为了能对内容文件或片段实现区分缓存替换，这里将副本服务器的缓存空间分成前缀片段区(PC，prefix cache)和后缀片段区(SC，suffix cache)。内容文件 C_k 的前缀片段表示为 $PrefixC_k$，它之后的其他内容片段统称为后缀片段。假设用户当前访问的是流媒体文件 C_k 的一个片段 $S_{k,i}$，且 $S_{k,i}$ 此时不在副本服务器缓存中，那么该片段是否应被缓存？若是，那么它应替换哪些当前缓存片段或内容文件？这 2 个问题就是 Diff-Attribute 缓存替换机制要回答的关键问题。副本服务器收到对片段 $S_{k,i}$ 的访问请求后，Diff-Attribute 缓存替换机制具体实现步骤如下。

1) 副本服务器收到对片段 $S_{k,i}$ 的访问请求后，首先判断 $S_{k,i}$ 所属内容文件 C_k 的前缀片段 $PrefixC_k$ 是否在 PC 缓存区中。

①若在，则执行第 4）步；

②若不在，则计算内容文件 C_k 的流行度分布熵

$H(C_k)$ 以及此时的熵临界值 Φ。

如果 $H(C_k) \geqslant \Phi$，则执行第 2)步；

如果 $H(C_k) < \Phi$，则执行第 4)步。

2) 副本服务器立即从源内容服务器获取内容文件 C_k 的前缀片段 $PrefixC_k$。

3) 判断 PC 缓存区可用空间是否足够缓存 $PrefixC_k$？

①若足够，则直接缓存；

②若不足，则可以将 PC 缓存区流行度最小的片段先移至 SC 缓存区。若 SC 区当前可用空间不足，则先将 SC 缓存区中流行度最小的片段删除，再将 PC 缓存区中流行度最小的前缀片段移至 SC 区。

4) 判断 SC 缓存区当前是否有足够的可用空间以缓存片段 $S_{k,i}$？

①若足够，则直接缓存 $S_{k,i}$；

②若不足，则分别计算片段 $S_{k,i}$ 和 SC 区中所有内容片段的流行度值。

5) 判断 SC 缓存区可用空间和流行度小于 $S_{k,i}$ 的片段所占空间的总和是否大于或等于 $S_{k,i}$。

①若是，则按照流行度值从小到大依次替换掉 SC 区中相应内容片段，直至 SC 缓存区当前可用空间能够缓存 $S_{k,i}$，再将 $S_{k,i}$ 缓存进 SC 区；

②否则，放弃缓存 $S_{k,i}$。

一个被缓存过的内容片段被替换出副本服务器之后，有可能又被请求访问。为此一旦某内容片段被替换出副本服务器，其相关的访问日志信息立即被清空，否则，该片段之前的访问流行度会干扰其他片段的缓存替换操作。当它被重新访问时，再为其建立新的访问日志。

3 Diff-Attribute 缓存替换机制性能评价

3.1 评价性能指标

为了评价 Diff-Attribute 的性能优劣，选取了 LRU、LFU、MRU、FIFO(first in first out)等典型缓存替换机制为比较对象，并从缓存命中率、字节命中率(即缓存内容有效命中率)、访问延迟启动率和传输成本消耗率等方面进行比较。这些性能指标具体定义如下：缓存命中率是指副本服务器中所有内容片段被访问的次数和副本服务器收到的总请求次数的比值；字节命中率是指用户从副本服务器中获得内容的字节数和用户请求内容的总字节数的比值；访问延迟启动率是指因用户所访问的文件的前缀片段没被缓存在副本服务器中而造成访问延迟启动的次数与总访问次数的比值；传输成本消耗率是指执行缓存替换机制所需的传输成本与没有缓存情况下所需要的传输成本的比值。

3.2 仿真环境设置

为了评估 Diff-Attribute 的缓存替换性能，设计了事件驱动的流媒体文件分发性能仿真系统，包括流媒体内容源服务器、边缘副本服务器和客户端 3 部分。为简化起见，假定内容源服务器与边缘副本服务器之间以及边缘副本服务器与客户端之间的传输链路有足够的带宽；另外，根据当前流行的互联网网间费用决算模型[13]，可以设定内容数据在这 2 类链路上的传输成本之比是 5:1。同时，为每个副本服务器配置一定缓存。为模拟用户内容请求和内容分发负载，主要实验参数的设置与文献[14]相同，如表 2 所示[14,15]。

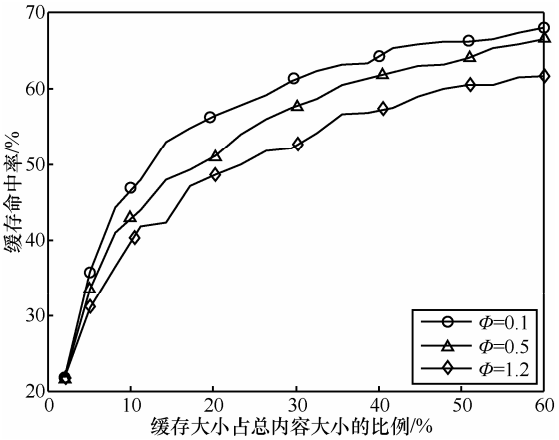
表 2 用户内容请求和内容分发负载模拟参数设置

参数名称	参数值
流媒体内容数量	100
前缀长度	5 min
流媒体内容大小范围	600~1 000 MB
流媒体内容访问概率	服从 θ=0.271 的 Zipf 分布
用户请求到达	服从 λ=0.1 次/秒的 Poisson 分布
用户请求个数	2 000

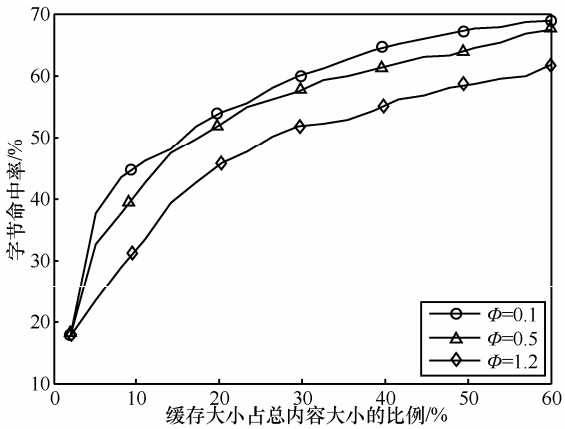
3.3 仿真结果与分析

熵临界值 Φ 的取值是 Diff-Attribute 缓存替换机制的关键。本文在分析不同缓存大小下，Φ 的 3 种取值策略对缓存命中率、字节命中率、访问延迟启动率和传输成本消耗率的影响。在某一时刻 T，根据内容访问的流行度分布情况和熵临界值 Φ，可得到副本服务器中内容文件集 D 流行度分布熵值为 0.5，内容文件最大流行度分布熵值为 1.2，内容文件最小流行度分布熵值为 0.1。

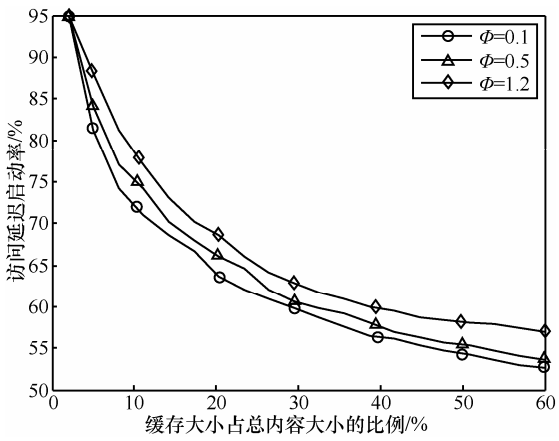
图 1 给出了副本服务器在不同配置缓存大小情形下，Φ 值对各性能指标的影响。图 1 分别对比了不同 Φ 值对缓存命中率、字节命中率、访问延迟启动率和传输成本消耗率的影响。由图 1 可知，Φ 值越小，缓存命中率和字节命中率越高，访问延迟启动率和传输成本消耗率越低。这是因为随着 Φ 值减小，$H(C_k)$ 大于 Φ 的内容文件数增多，更多内容文件的前缀片段将缓存在副本服务器 PC 缓存区，致使用户直接向内容源服务器的内容请求数减少，副



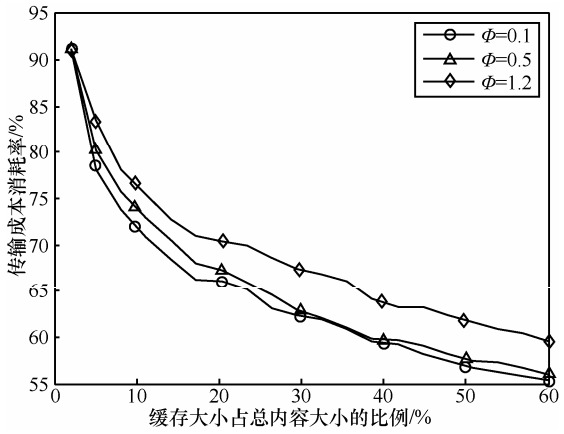
(a) Φ 取值策略对缓存命中率的影响



(b) Φ 取值策略对字节命中率的影响



(c) Φ 取值策略对访问延迟启动率的影响



(d) Φ 取值策略对传输成本消耗率的影响

图 1　不同缓存大小情形下，不同 Φ 取值策略对各性能指标的影响

本服务器缓存命中率和字节命中率增大，进而访问延迟启动率和传输成本消耗率降低。但是当 Φ 取最小熵值时，缓存替换开销将增大，因为此时任何片段被请求访问时都要先处理其所属内容文件的前缀片段的缓存，然而并不是所有的前缀片段都具有较高的流行度，特别是当 PC 区已满后，Diff-Attribute 将用 PC 区中流行度低的前缀片段替换 SC 区中流行度较高的片段。这不仅增大缓存替换开销，而且缓存命中率也不会改善。当 Φ 取整体流行度分布熵值时，Diff-Attribute 的缓存性能与 Φ 取最小熵值时类似。但当 Φ 取最大熵值时，其缓存性能远远落后于前两者。这是因为当 Φ 取最大熵值时，Diff-Attribute 只依据各片段的流行度 $H(S_{k,i})$ 大小进行缓存替换，不能提前缓存热门内容文件的前缀片段，导致缓存替换性能较差。由此可知，Φ 取整体流行度分布熵值能获取更好的缓存替换性能。

图 2 和图 3 分别给出了 Diff-Attribute 与其他典型机制在缓存命中率和字节命中率等方面的性能对比。随着副本服务器配置缓存空间的增大，副本服务器中缓存的内容数量增多，命中率必定增大。可以看出 Diff-Attribute 比 LFU 的缓存命中率和字节命中率平均高出 6%和 8%。因为按式(6)的熵临界值策略，Diff-Attribute 可选出热门内容文件的前

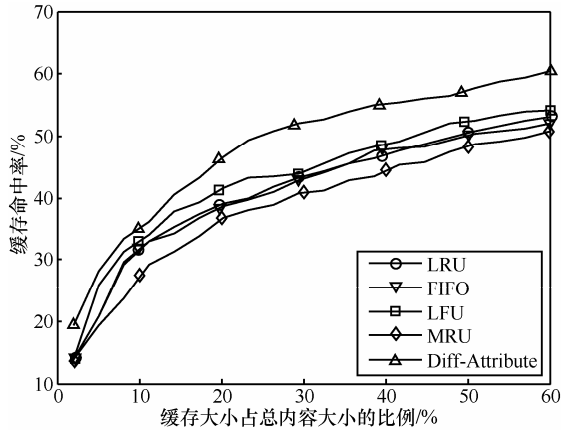


图 2　不同缓存大小下的缓存命中率

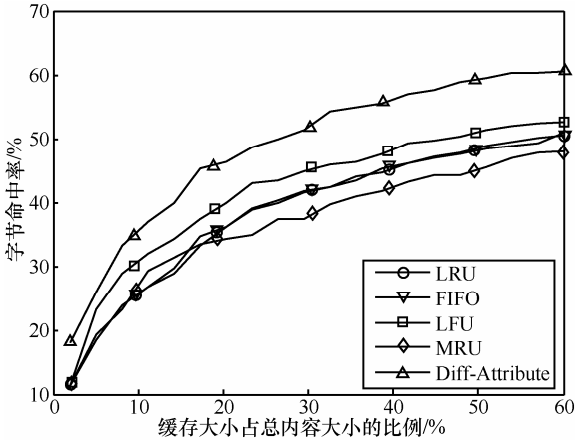


图 3　不同缓存大小下的字节命中率

缀片段，将其提前缓存在副本服务器中，从而有效提高了缓存命中率和字节命中率。

图 4 给出了 Diff-Attribute 与其他 4 种机制在访问延迟启动率的比较。由图 4 可知，Diff-Attribute 的访问延迟启动率一直低于其他 4 种算法，而且分别比 MRU、FIFO、LRU 和 LFU 最大降低 18%、15%、14%和 13%。究其原因在于：Diff-Attribute 能够提前缓存更多前缀片段，使用户请求访问的初始部分内容增多，因此在降低访问延迟启动率方面有较大优势。

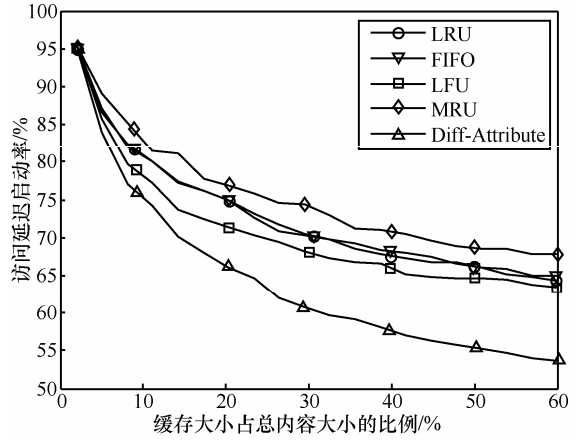


图 4　不同缓存大小下的访问延迟启动率

图 5 给出了副本服务器不同配置缓存大小下，Diff-Attribute 与其他 4 种算法在传输成本消耗率上的比较。一般而言，若副本服务器配置的缓存空间越大，副本服务器中的缓存内容就越多，则用户向源内容服务器的内容请求数就越少，从而使它们的传输成本消耗逐渐降低。但相对而言，Diff-Attribute 的传输成本远低于其他 4 种算法。究其原因在于：Diff-Attribute 能够提前缓存热门内容前缀，减少了向源内容服务器的请求次数，使其在内容提供商到副本服务器之间的传输成本降低，而此段比副本服务器到客户端的传输成本高，因此其整体传输成本消耗率相对更低。

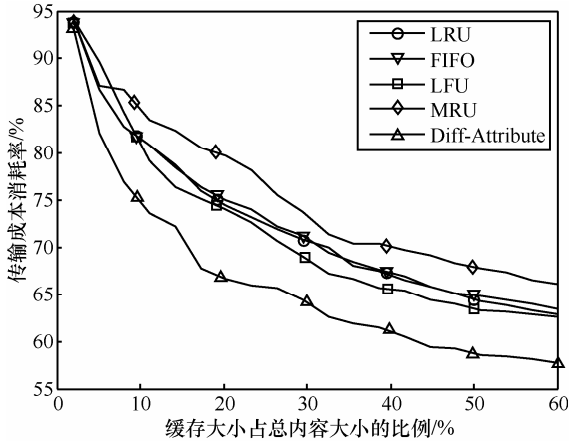


图 5　不同缓存大小下的传输成本消耗率

4　结束语

在 CDN-P2P 融合内容分发网络中，副本服务器缓存替换机制的高效与否，对分发性能影响很大。实际上，流媒体文件的访问流行度呈现较强的非均衡性，不仅体现在各文件的整体访问流行度按 Zipf 分布，而且一个流媒体文件的各个内容片段的访问流行度也不相同。然而，现有机制仅关注内容文件的整体流行度，而没有考虑其中各片段的个体流行度差异性，尤其在内容片段被访问次数和片段所在内容文件的访问流行度相同情形下，现有机制无法区分哪些缓存片段该被替换。因此现有机制无法提高预缓存内容片段的访问命中率，并降低用户内容访问延迟。为此本文以内容文件或内容片段访问流行度为决策依据，提出了基于内容访问属性差异性的缓存替换机制 Diff-Attribute。

该机制突出的特点在于：一是既考虑了内容文件的整体流行度，也考虑了内容文件中各个片段的个体流行度；二是基于分布熵，定义了一种内容流行度均衡性度量方法和基于整体流行度分布上的熵临界值取值策略。若流行度均衡，则提前缓存各文件的前缀片段；若不均衡，则提前缓存热门文件或其中最热门的内容片段。仿真结果表明，与其他 2 种取值策略（最大熵值、最小熵值）相比，整体流行度分布熵策略具有较好的缓存替换性能；在缓存命中率和字节命中率方面，Diff-Attribute 机制分

别高出传统机制（如 LFU、LRU、MRU、FIFO 等）约 6%和 8%；在访问延迟启动率和传输成本消耗率方面，Diff-Attribute 机制则降低了约 13%和 7%。

参考文献：

[1] CAROFIGLIO G, MORABITO G, MUSCARIELLO L, *et al*. From content delivery today to information centric networking [J]. Computer Networks, 2013, 57(16): 3116-3127.

[2] HAMMAMI C, JEMILI I, GAZDAR A, *et al*. Hybrid live P2P streaming protocol[J]. Procedia Computer Science, 2014, 32: 158-165.

[3] GARMEHI M, ANALOUI M, PATHAN M, *et al*. An economic replica placement mechanism for streaming content distribution in hybrid CDN-P2P networks[J]. Computer Communications, 2014, 52(1): 60-70.

[4] FAMAEY J, ITERBEKE F, WAUTERS T, *et al*. Towards a predictive cache replacement strategy for multimedia content [J]. Journal of Network and Computer Applications, 2013, 36(1):219-227.

[5] GUAN N, YU M, YI W. WCET Analysis with MRU caches: challenging LRU for predictability[A]. Proceedings of the 18th IEEE Real Time and Embedded Technology and Applications Symposium[C]. Beijing, China, 2012.55-64.

[6] GALLO M, KAUFFMANN B, MUSCARIELLO L, *et al*. Performance evaluation of the random replacement policy for networks of caches [J]. Performance Evaluation, 2014, 72(1):16-36.

[7] YANG G, LIAO J, ZHU X. Proxy caching algorithm based on segment popularity for mobile streaming media [J]. Journal on Communications, 2007, 28(2): 33-39.

[8] WU T, KOEN D, WERNER V. Reuse time based caching policy for video streaming[A]. Proceedings of the 9th Annual IEEE Consumer Communications and Networking Conference-Multimedia & Entertainment Networking and Services[C]. Las Vegas USA, 2012. 89-93.

[9] SALEH O, HEFEEDA M. Modeling and caching of peer-to-peer traffic[A]. Proceedings of the 14th IEEE International Conference on Network Protocols[C]. Santa Barbara USA, 2006.249-258.

[10] CHOI J, REAZ A, MUKHERJEE B. A survey of user behavior in VoD service and bandwidth-saving multicast streaming schemes[J]. IEEE Communications Surveys & Tutorials, 2012, 14(1): 156-169.

[11] LIU Y, LI F, GUO L. A server's perspective of internet streaming delivery to mobile devices[A]. Proceedings of the 24th IEEE INFOCOM[C]. Orlando USA, 2012.1332-1340.

[12] Study uncovers critical link between video quality and audience retention, revenue opportunities[EB/OL]. http://www.akamai.com/, 2013.

[13] ZHANG N, LEVA T, HAMMAINEN H. Value networks and two-sided markets of internet content delivery [J]. Telecommunications Policy, 2014, 38(5): 460-472

[14] KUO J, SHIH C, HO C, *et al*. A cross-layer approach for real-time multimedia streaming on wireless peer-to-peer ad hoc network[J].Ad Hoc Networks, 2013, 11(1):339-354.

[15] OH H, SONG H. Metafile-based scalable caching and dynamic replacing algorithms for multiple videos over quality-of-service networks [J]. IEEE Transactions on Multimedia, 2007, 9(7): 1535-1542.

作者简介：

聂华（1972-），男，山东济南人，北京科技大学高级工程师，主要研究方向为互联网与云数据中心。

张敏（1972-），女，重庆人，北京科技大学副教授，主要研究方向为互联网理论与技术。

郭敬荣（1986-），女，河北衡水人，北京科技大学硕士生，主要研究方向为互联网理论与技术。

阳小龙（1970-），男，四川邻水人，北京科技大学教授、博士生导师，主要研究方向为互联网理论与技术。