

流媒体代理缓存技术研究^{*}

余江¹ 刘威¹ 王泰¹ CHOU Chun-Tung²

(华中科技大学电子与信息工程系 武汉 430074)¹ (新南威尔士州大学计算机科学和工程学院 澳大利亚)²

摘要 随着流媒体技术在互联网上的广泛应用,原用于加速 Web 内容分发的代理缓存技术也被应用于流媒体内容分发领域。然而,由于流媒体对象的一些独特特点,需要在传统 Web 缓存技术的基础上提出适用于流媒体对象的缓存技术。本文分析了基于代理缓存的流媒体分发中的关键问题和技术挑战,对现有的各种流媒体缓存方案进行了综述、分类和比较,最后对流媒体代理缓存的未来发展方向进行了展望。

关键词 流媒体,代理缓存,Web 内容分发

Survey on Proxy Caching Technologies for Streaming Media

YU Jiang¹ LIU Wei¹ WANG Tai¹ CHOU Chun-Tung²

(Department of Electronics and Information Engineering, Huazhong University of Science and Technology, Wuhan 430074)¹

(School of Computer Science and Engineering, University of New South Wales, Australia)²

Abstract With the widespread uses of the streaming technology over Internet, proxy caching has been introduced to the area of media streaming from that of Web content distribution. Due to some distinct characteristics of streaming media object, novel caching techniques for streaming media are required to replace the traditional Web caching techniques. The critical issues and challenges of proxy caching strategies for streaming media are reviewed in this paper. We survey, classify, and compare the state-of-art various proxy caching solutions for streaming media. Finally, the future research issues in this field are outlined.

Keywords Streaming media, Proxy caching, Web content distribution

1 引言

近年来,随着多媒体点播应用的日益普及,多媒体内容的传输占用了当前 Internet 上的大部分流量,导致骨干带宽紧张和用户接入时延增加。代理缓存正是缓解这一矛盾的有效技术手段。该技术通过将代理服务器部署在网络边缘靠近客户的地方,对热点对象进行缓存,使后续的视频流媒体请求由缓存提供服务来达到减轻骨干网络和服务器负载,提高用户服务质量的目的。具体网络拓扑如图 1 所示。

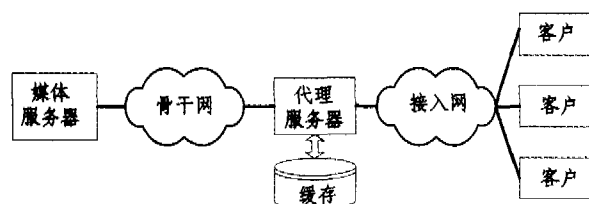


图 1 部署代理缓存的网络体系结构

代理缓存技术最初是用在 Web 内容的分发上。然而研究发现,流媒体内容在多个方面呈现出与普通 Web 内容不同的特征。第一,流媒体需要占用比普通 Web 内容更大的存储空间,如果对流媒体对象进行类似于 Web 一样的整体缓存,少数视频流媒体对象就可以消耗掉所有的缓存容量,将大大

降低缓存的利用率和命中率。第二,流媒体的分发需要在一段时间持续占用巨大的磁盘 I/O 带宽和网络带宽,以网络为主要研究场景时降低带宽的消耗成为主要考虑的技术目标之一。第三,用户通常会在流媒体播放过程中进行交互式操作,可能导致同一个会话中的流接入速率在不同时刻有所不同,对缓存的管理带来困难。第四,与无播放质量要求的传统 Web 内容不同,用户对于流媒体内容有一定的播放质量(如响应时延、画面抖动)需求,需要对从代理服务器播放的视频流和从媒体服务器播放的视频流采用一定的调度策略。

如何针对流媒体对象的特点研制新型缓存机制,是目前多媒体分发技术发展所迫切需要解决的问题。本文对近年来提出的流媒体代理缓存技术的各类方法进行了全面综述和分析。后文组织如下:第 2 节对现有的代理缓存技术进行综述,第 3 节对各种算法进行分类分析和比较,最后对代理缓存技术的发展趋势进行展望。

2 几种典型的流媒体缓存算法

2.1 选择缓存算法

在用户接入带宽资源有限的情况下,保证正常的媒体播放速率成为媒体缓存算法的主要设计目标。有一类缓存算法,根据带宽情况对影响媒体播放的关键数据或者内容有选择地缓存,我们把这类缓存算法称为选择缓存算法(Selective

^{*}基金资助:国家自然科学基金(No. 60302004),澳大利亚研究院国际合作基金(LX0240468)。余江 博士研究生,研究方向为多媒体内容分发技术;刘威 讲师,研究方向为多媒体内容分发技术;王泰 博士研究生,研究方向为多媒体内容分发技术;CHOU Chun-Tung 博士,Senior Lecturer,研究方向为流量工程,内容分发网络等。

Caching)。

Miao 等人以保证正常媒体播放为前提,在考虑磁盘容量的约束条件下,提出了两种媒体数据帧选取缓存算法(Frame-Selected Caching)^[1],分别针对骨干链路有无 QoS 保证时的情形。当骨干链路有 QoS 保证时,帧选取的原则是使得骨干链路的带宽需求和客户端播放缓存需求最小;当骨干链路无 QoS 保证时,帧选取的原则是使得代理缓存可以为客户端播放缓存提供尽可能多的帧,以提高网络拥塞时媒体播放的鲁棒性。

同样以降低骨干链路带宽需求为目标,Zhang 提出不以媒体数据帧为缓存对象,而以部分超速率数据为对象的速率分段缓存算法(Rate-Staged Caching)^[2]。该算法针对 VBR 流式传输,对高于指定数据传输阈值的媒体数据部分进行缓存。类比于时域上的前缀缓存,速率分段缓存算法也存在如何选取截断速率的问题。文[2]的研究表明,在磁盘容量和网络带宽两个约束下,确定优化的速率是一个背包问题。

2.2 基于间隔的缓存算法

用户请求具有高度时域邻近性(Temporal locality)的情况下,对同一媒体对象的两个连续播放请求的间隔内容做缓存,可以显著降低服务器的吞吐量。该类缓存算法称为基于间隔的缓存(Interval Caching),最初的思想由 Dan 提出^[3]。当存在对媒体对象相同部分的两个连续请求时,则连续请求的间隔部分将随着播放过程被缓存。该算法目前已经有多种衍生版本,最具代表性的是 Tewari 提出的基于资源的缓存(Resource based caching, RBC)算法^[4]。RBC 算法以每个媒体对象对缓存空间和传输带宽的需求为约束条件,提出了一种启发式的算法选择媒体对象的缓存粒度进行选择,其粒度可以是一个滑动间隔、一系列邻近间隔的组合(文[4]称为游程,run),甚至是完整的媒体对象。

2.3 基于分段的缓存

基于间隔的缓存算法,其有效性依赖于用户请求在时域上的有效性。当到来请求的时间间隔较长时,其性能有可能退化到全对象缓存。为此,研究人员开始考虑选取部分的媒体关键内容进行缓存,例如由 Sen 等提出的前缀缓存(Prefix Caching)^[5],是将媒体内容分成前缀(Prefix)和后缀(Suffix)两个片段部分,并且优先缓存前缀。前缀缓存可以有效地降低用户播放媒体对象时的启动时延,是目前应用较为广泛的流媒体缓存技术。其不足之处在于媒体前缀部分的长度不容易确定。Jin 等人的研究^[8]表明,在有限的网络资源的约束下,为达到指定的启动时延的前缀缓存,其部署是一个部分背包问题(Fractional Knapsack)。

更为普遍的方式是将媒体内容分为多个片段,即基于分段的缓存(Segment based Caching,以下简称分段缓存)。分段缓存将媒体内容沿着播放时间分成多个片段(Segment),并将片段作为存储和置换的基本单元。分段策略是分段缓存的主要研究热点,依据它可以将分段缓存算法分成等长划分和变长划分两类。变长划分算法的代表是 Wu 等人^[6]提出的片段长度以指数增长的缓存算法策略,这类策略设基本的存储单位为块(block),则第 i 个片段由第 2^{i-1} , $2^{i-1}+1$, ..., 2^i-1 块构成,片段长度为 2^{i-1} 个块。由于可以通过一次性丢弃较大的后续片段,来适应缓存流行度的变化,该算法可以取得比普通的前缀缓存更低的平均启动时延和更高的字节命中率。在等长划分算法中,片段长度的选择是关键,文[7]提出基于媒体的受访特性确定等分片段的长度,可以获得比指数分段

缓存或者前缀缓存更低的网络带宽占用^[8]。分段缓存算法可以提供较为精细的缓存粒度,带来更多的灵活性,成为目前主流的代理服务器缓存算法之一。

2.4 结合编码转换的缓存算法

然而,随着网络使用的普遍化,用户接入网络呈现出多样化的发展趋势,用户的终端环境、接入带宽以及允许获得的服务质量各不相同,因此请求的视频质量也可能各不相同。为了适应接入网络带宽的不同需求,需要根据用户的请求和带宽状况,动态地改变缓存内容质量和传输速率。结合编码转换技术,其缓存算法可以利用向不同用户提供不同版本的媒体内容来满足用户的需求,有两种简单方案:(1)缓存媒体的完整版本(Full Version Only, FVO),每次根据用户请求由代理服务器进行编码转换。这种方式对代理服务器的 CPU 资源提出了很高的要求,无法很好地为不同的网络用户服务。(2)仅缓存经过编码转换后的流媒体(Transcoded Version Only, TVO),编码转换预先由媒体服务器完成。这样的存储方式难以满足所有用户对流媒体质量的不同要求,而且会对网络带宽带来较大压力。

为了解决这两种方案之间的矛盾,Xueyan Tang 等人提出了一种自适应的缓存方式^[9]。该算法根据代理服务器 CPU 资源和网络带宽资源的状况,动态地决定是缓存原始版本还是缓存编码转换后的版本。这种方法的缺点在于可能会对同一个视频对象缓存多个版本,对缓存空间提出了较高的要求。

Masahiro^[10]等人则建议只为每个媒体对象缓存所得到的最高质量的版本,但是代理服务器可以灵活地采用最大申请(根据网络带宽)、平均申请(根据一段时间内用户申请质量的最大值与网络带宽比较,取较小值)和等同申请(按照用户请求的质量申请)三种不同的策略向媒体服务器请求所需内容。这种算法在缓存空间受限的情况下对缓存效率改善明显。然而,由于代理缓存可向服务器申请不同质量的流媒体版本,媒体服务器仍然必须为此承担编码的负载。

为了降低媒体服务器的编码负担,Bo Shen 等人^[11]提出了一种可编码转换的缓存系统,采用降低比特速率(bit-rate reduction)和降低空间分辨率(spatial resolution reduction)两种编码转换技术在网络边缘完成质量自适应。根据缓存版本数量和缓存命中后的行为,文献提出了 TEC-11、TEC-12 和 TEC-2 三种缓存策略。TEC-11 和 TEC-12 都只会对同一个缓存对象存储一个版本,不同的是 TEC-11 总是缓存最高质量的,TEC-12 总是缓存最低质量的,而 TEC-2 则会为同一个缓存对象存储多个版本。

随着编码理论和信号处理技术的发展,部分编码转换技术几乎可以达到实时的效果^[20],大大增加了此类技术的实用性。

2.5 结合可扩展编码的缓存算法

结合可扩展编码的缓存算法同结合编码转换技术的缓存算法一样,可以很好地满足不同接入速率要求,但是它利用的是缓存对象采用的可扩展编码的属性。可扩展编码通常也称为分层编码,采用分层的比特流结构对媒体数据进行压缩。第 1 层称为基本层,可以被独立解码;其余层称为增强层,其解码依赖于基本层;获得的数据层次越多,解码出来的质量越高,所占用的带宽也越大^[12]。结合可扩展编码的缓存技术的主要思路是通过向不同的客户提供编码数据的不同层次子集来达到质量自适应。

Jussi Kangasharju 首先提出了以层为单位 Revenue 缓存算法^[13]。该算法以网络带宽和缓存空间为约束,以总体缓存受益为优化目标,将层次作为基本粒度,一旦满足条件,视频对象的一层都将整体被缓存。缓存层次的选择建模为一个随机背包问题,采用启发式算法来决定为每个视频选择什么数量的缓存层次。该算法充分考虑了约束条件对缓存效率的影响,但是其以整层作为缓存粒度不够精细,缓存空间的利用率有待提高。

Reza Rejaie 等人则将每一层划分为等长片段^[14],当缓存内容的质量高于用户的请求(即所缓存的层数高于用户请求

的层数)时,代理服务缓存只需要向媒体服务器预取那些同层次或低层次缺失的片段数据。反之,如果缓存内容的质量低于用户的请求,则代理服务缓存还需要对缺失的高层次数据进行预取。

Stefan Podlipnig 等人^[15]进一步改进了可扩展编码算法中的置换策略部分,提出了一个新的性能评价指标——质量权重命中率,用来衡量请求命中率与缓存质量之间的平衡程度。并在原有水平和垂直的置换策略的基础上,提出了改进的垂直与水平相结合的置换策略,如图 2 所示。

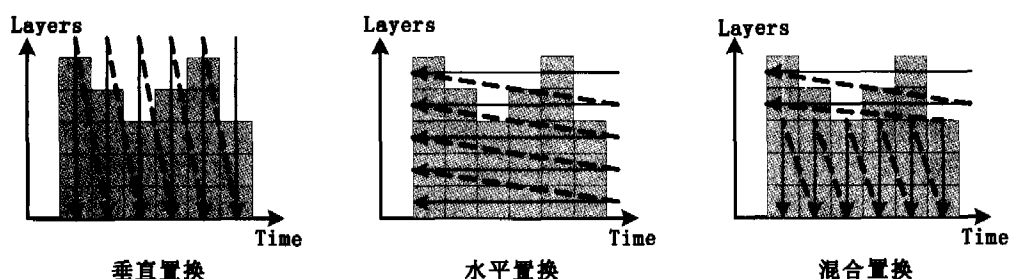


图 2 置换策略^[15]

实验发现,垂直与水平相结合的置换策略可以在改善字节命中率(byte-hit-rate, BHR)与提高质量命中率(quality-hit-rate, QHIT)之间取得较好的平衡且有良好的适应性。

M. Zink 等人则对可扩展编码缓存算法中的预取算法进行了研究,提出一种无窗口大小限制的预取算法,该算法通过预取消除同一层次的裂缝来达到降低缓存内容层次波动的目的。他们提出的 Polishing 算法^[16]从质量变化频率和幅度两个方面对缓存内容进行了描述,缓存算法的目标是使谱函数值最小的情况下,将尽可能多的缓存片段传送给用户。

随着可扩展编码技术的发展,Jiangchun Liu 等人针对采用 MPEG-4 FGS 编码的媒体提出了 FGS 缓存算法^[17]。该算法在粒度划分上进一步克服了原有算法不够精细的缺点,可以将视频对象以任意速率进行传输。该算法以在骨干链路带宽消耗和媒体传输质量之间获得最优折衷为目标,设计了一个良好的缓存管理框架,但预取和置换方案还有待讨论和实

现。

3 各种缓存算法的分析和比较

根据是否考虑代理服务器和客户端之间网络带宽的差异性,本文将目前的流媒体缓存技术分为针对同构网络用户和针对异构网络用户的两类代理缓存技术,下面我们分别对其进行了分析和比较。

上文提到的选择缓存算法、基于间隔的缓存算法和基于分段的缓存算法均属于针对同构网络用户的范畴。这类算法中,代理服务器所服务的用户终端一般具有相同或者类似的处理能力和接入带宽,缓存算法的设计难点在于媒体数据缓存部分的选取。我们对其中具有代表性的算法从实现复杂度、算法扩展性以及及时延和带宽方面的性能进行了比较,如表 1 所示。

表 1 针对同构网络用户的代理缓存技术比较

算法类别	算法名称	缓存数据的选取	算法复杂度	算法扩展性	时延性能改进	带宽节省改进
选择缓存	Frame-Selected Caching[1]	部分数据帧	高	高	中等	高
	Rate-Staged Caching[2]	超出速率阈值的数据	中等	中等, 速率阈值的选取与场景有关	中等	高
基于间隔的缓存	Interval Caching[3]	滑动间隔	低	低, 在请求到来时域稀疏时性能低	高	中等
	Resource based Caching[4]	滑动间隔/滑动间隔的组合/整个媒体	中等	高	高	中等
基于分段的缓存	Prefix Caching[5]	媒体初始前缀数据	低	中等, 前缀长度的选取与场景有关	高	中等
	Exponential Segment Caching[6]	长度指数增加的媒体片段	中等	高	高	中等
	Adaptive and Lazy Caching[7]	长度相等的媒体片段	高	高	高	高

由表 1 可见,这类算法普遍存在两方面的矛盾。一方面,代理缓存算法存在低复杂度和高扩展性之间的矛盾。例如,基于间隔的缓存中,Interval Caching 易于实现,但是其扩展性有限,无法对时间临近度较低的用户请求做出反应;RBC

提出的多种粒度的间隔缓存提高了算法的扩展性,但是实现复杂度也相应增加。又如选择缓存中,Rate-Staged 缓存易于实现,但是其算法扩展性受到阈值选取优化问题的困扰;Frame-Selected 缓存提出的在网络服务质量和缓存容量的约

束条件下的帧选取算法具有较高的扩展性,但其实现复杂度也相应提高。比较而言,基于片段的缓存算法的复杂度和扩展性适中,因此获得了广泛的研究者关注。

另一方面,代理缓存算法性能在降低用户体验时延和节省骨干链路带宽的两个性能之间存在着折衷。例如,基于间隔的缓存算法和选择型缓存算法分别以两个性能之一作为各自主要的性能考量,却对另一个性能的改进不明显。对基于片段的缓存算法而言,由于片段划分策略通常支持前缀缓存,因此在降低用户启动时延方面有明显优势。片段缓存的性能

在这两个性能之间也存在着明显的权衡关系^[11]。

结合编码转换和结合可扩展编码的缓存技术均属于针对异构网络用户的范畴。然而两者的实现机制和设计难点均不一样。结合编码转换的缓存技术主要通过传送不同编码速率版本的视频内容来为不同需求的用户服务,其研究内容集中在如何对不同版本进行缓存,从而在 CPU 资源、网络带宽和缓存空间之间取得良好的平衡。我们对这类技术的典型算法从缓存版本数量、质量依据、资源消耗等方面进行了比较,具体分析如表 2 所示。

表 2 结合编码转换技术的缓存方案比较

算法名称	策略名称	缓存质量依据	同目标缓存版本数量	编码转换的位置	对异构网络的支持	资源需求	
						CPU	存储空间
Adaptive Caching[9]	Adaptive Caching	S-P 带宽和 CPU 资源共同决定	多个	代理缓存	高	中	中
Quality Adaptive Caching[10]	QA (max)	S-P 带宽	1 个最高质量	原始服务器/代理缓存	中	高	低
	QA (avg)	C-P 和 S-P 带宽大小	1 个最高质量	原始服务器/代理缓存	中	高	低
	QA (same)	C-P 带宽大小	1 个最高质量	原始服务器/代理缓存	中	高	低
Transcoding Enabled Caching[11]	TEC-11	C-P 带宽大小	1 个最高质量	代理缓存	中	高	低
	TEC-12	C-P 带宽大小	1 个最低质量	代理缓存	中	低	低
	TEC-2	C-P 带宽大小	多个	代理缓存	高	中	高

而结合可扩展编码的缓存技术则是通过向用户提供不同编码层次集合来满足用户不同接入速率要求,不同粒度下的预取和置换方案是其主要的研究内容。我们对目前该技术采

用的典型方案从缓存粒度、预取和置换算法、缓存目标等方面进行了比较,结果如表 3 所示。

表 3 可扩展编码缓存技术方案比较

算法名称	缓存粒度	预取方案	置换顺序	目标
Revenue Cache[13]	层	——	垂直置换	节省 S-P 带宽
Segment-grained layering Cache[14]	片段	窗口式	水平置换	提高缓存内容的连续性和完整性
Combined replacement Cache[15]	片段	窗口式	水平与垂直相结合	在字节命中率和质量命中率间取得最佳平衡
Polishing Cache[16]	片段	无窗口大小限制	水平置换	降低缓存内容质量变化的幅度和频率
FGS Cache[17]	任意大小	——	——	节省 S-P 带宽, 提高发送质量

可扩展编码技术可以通过发送不同层次的子集来进行灵活的质量自适应,具有缓存空间需求小等优点;但是该技术也具有对包丢失敏感、同等质量下比单一层次编码需要更多带宽等缺点。编码转换技术则可以拥有良好的容错性,但是对于缓存系统,该技术对缓存空间提出了较高的要求。

由于结合可扩展编码的缓存技术和结合编码转换的缓存技术都具有各自的优势,在不同的比较场景下的比较结果往往不同。

文[18]在较简单的情况下对异构网络环境下代理缓存技术进行了性能的比较,它将结合编码转换的缓存策略分为仅缓存高质量或仅缓存低质量两个策略,分层编码采用 2 层编码。仿真结果表明:高质量版本缓存总是可以得到最好的视频质量,但是需要付出较高的网络和时延代价;低质量版本缓存虽然拥有较低的网络负载和播放时延,但是始终无法得到较好的视频质量;只有采用分层编码的缓存技术在这三者当中取得了较好的平衡。文[17]考虑 5 种质量版本,采用 MPEG-4 精细粒度编码作为可扩展编码技术,从骨干网带宽

消耗和代理服务器计算效率两个方面对两种缓存技术进行了深入比较,进一步验证了结合可扩展编码缓存技术的优越性。

但是,采用分层编码,为达到同样的视频质量,需要比单一层次编码占用更多的存储空间,在这个额外负载很大的情况下,将极大降低采用分层编码缓存技术的效率。文[28]针对该问题,提出只有采用可扩展编码和编码转换技术的混合缓存策略,才可以很好地应对额外负载大小变化对缓存效率的影响。而文[19]在一个易丢包的环境下再次对上述两种技术进行比较,发现在中等的包丢失率和骨干链路带宽足够的环境下,结合编码转换技术的缓存策略是最佳选择。

因此,异构网络环境下采用哪一种缓存策略,需要结合代理服务器的使用环境、用户特点等诸多方面加以综合考虑。

结论与展望 代理缓存可以有效降低流媒体分发时的用户体验时延和骨干链路带宽需求。本文根据代理服务器和用户终端之间用户接入网络情况的不同,从同构网络用户和异构网络用户两种情况,对现有的各种流媒体代理缓存策略进

(下转第 44 页)

当一个网格用户通过网格入口访问网格系统中的资源时,如果通过了网格认证服务的认证,认证服务就会和相关的网格节点联系,网格节点的访问控制服务就会赋予用户一系列角色,同时认证服务在用户机器上建立上下文 Agent,用来获取与用户相关的上下文。同样,在网格节点上,建立上下文 Agent,用来获取共享资源的上下文。这些 Agent 能动态地从应用环境中获取上下文,把传统的网格计算环境转换成了有感知功能的环境。当上下文变化时,访问结果也应不同。

结论 为满足网格应用中复杂的访问控制需求,本文提出动态的基于角色和上下文的网格访问控制模型,该模型具有以下优点:①扩展了 RBAC 模型,加入了上下文约束,具有 RBAC 模型的优点。②该模型能动态地实施,具有很高的灵活性。在设计时,管理员能规定复杂的上下文敏感的授权策略。在运行时,基于动态变化的上下文,计算上下文约束条件,授权服务能自动实施上下文敏感的授权策略。③该模型易于扩展。上下文信息与应用程序分离,上下文的类型定义与访问控制规则无关,改变上下文的类型定义不会影响应用程序。

目前,RCBAC 模型已经在实验网格应用平台中作为核心的授权与访问控制模块。在下一步研究工作中拟讨论上下文约束的引入对 RBAC 模型的角色约束和角色层次的影响,从而以完整的基于角色和上下文的访问控制机制实现用户自定义的网格访问控制。

参考文献

- 1 Foster I, Kesselman C, Tuecke S. The Anatomy of the Grid;

- Enabling Scalable Virtual Organizations. *International Journal of High Performance Computing Applications*, 2001, 15(3): 200~222
- 2 Foster I, Kesselman C, Tuecke G, et al. A Security Architecture for Computational Grids. In: *ACM Conference on Computer and Communications Security Conference*, 1998. 83~91
- 3 郝志辉, 陈渝, 刘鹏. 网格计算[M]. 北京:清华大学出版社, 2002. 67~80
- 4 Pearlman L, Welch V, Foster I, et al. A Community Authorization Service for Group Collaboration. In: *Proceedings of the IEEE 3rd International Workshop on Policies for Distributed Systems and Networks*, 2002
- 5 Alfieri R, Cecchini R, et al. VOMS, an Authorization System for Virtual Organizations. <http://gridauth.infn.it/docs/VOMS-Santiago.pdf>
- 6 Johnston W, Mudumbai S, Thompson M. Authorization and Attribute Certificates for Widely Distributed Access Control. In: *Proceedings of IEEE 7th International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises*, 1998
- 7 Chadwick D W, Otenko A. The PERMIS X. 509 role-based privilege management infrastructure. In: *Proceedings of the Seventh ACM Symposium on Access Control Models and Technologies*, Monterey, California, USA. 2002
- 8 Sandhu R, Coyne E, Feinstein H, et al. Role-Based Access Control Models. *IEEE Computer*, 1996, 29(2): 38~47
- 9 Mostéfaoui G K, Brézillon P. A Generic Framework for Context-Based Distributed Authorizations. In: *Proc. 4th International and Interdisciplinary Conference on Modeling and Using Context (Context'03)*, LNAI 2680, 2003. 204~217
- 10 McDaniel P. On Context in Authorization Policy. In: *Proceedings of the Eighth ACM Symposium on Access Control Models and Technologies*, 2003
- 11 Deloach S A, Wood M F, Sparkman C H. Multiagent Systems. *Int'l J Software Eng and Knowledge Eng*, 2001, 11(3): 231~258

(上接第 21 页)

行了分类比较,重点介绍了不同缓存策略面临的问题、解决的思路 and 局限性。

流媒体代理缓存技术除了自身仍然面临着众多的选择和折衷,在其他很多方向上还有很多问题有待解决。

首先,随着无线网络的不断发展,针对无线信道传输可靠性差、误码率高、用户移动等特点,建立有效的缓存策略,使得部署在传统 Internet 与无线网络边缘的代理服务器为无线用户提供高效的流媒体服务,是目前研究的一个热点。

其次,现有的缓存策略的研究都是将 Web 对象与流媒体对象分开考虑,这与大多数实际情况不符,如何建立高效的混合缓存策略也是需要研究的问题。

最后,合作式缓存作为提高缓存系统扩展性的一个方案得到了广泛关注,设计一个流媒体对象在合作式缓存系统中进行高效分布和缓存的方案,对提高流媒体缓存技术的扩展性具有重要意义。

参考文献

- 1 Miao Zhou, Ortega A. Scalable proxy caching of video under storage constraints. *IEEE JSAC*, 2002, 20(7): 1315~1327
- 2 Zhang Z-L, Wang Y, Du D H C, et al. Video staging: A proxy-server-based approach to end-to-end video delivery over wide-area networks. *IEEE Trans Networking*, 2000, 8: 429~442
- 3 Dan A, Sitaram D. A generalized interval caching policy for mixed interactive and long video workloads. In: *Proc. IS&T/SPIE Conf on Multimedia Computing and Networking (MMCN'96)*, 1996
- 4 Tewari R, Vin H, Dan A, et al. Resource-based caching for web servers. In: *Proc. SPIE/ACM Conf on Multimedia Computing and Networking (MMCN'98)*, 1998
- 5 Sen S, Rexford J, Towsley D. Proxy prefix caching for multimedia streams. In: *Proc. IEEE INFOCOM'99*, 1999
- 6 Wu K, Yu P, Wolf J. Segment-based proxy caching of multimedia streams. In: *Proc. 10th Int'l World Wide Web Conf. (WWW'01)*, 2001

- 7 Chen S, Shen B, Wee S, et al. Adaptive and lazy segmentation based proxy caching for streaming media delivery. In: *Proc. NOSS-DAV'03*, 2003
- 8 Chen Songqing, Shen Bo, Wee Susie, et al. Investigating Performance Insights of Segments-based Proxy Caching of Streaming Media Strategies. In: *Proceedings of SPIE/ACM International Conference on Multimedia Computing and Networking (MMCN'04)*, 2004
- 9 Tang Xueyan, Zhang Fan, Chanson S T. Streaming Media Caching Algorithms for Transcoding Proxies. In: *Proc. ICPP 2002*, 2002
- 10 Sasabe M, Taniguchi Y, Murata M, et al. Proxy Caching Mechanisms with Quality Adjustment for Video Streaming Services. *IEEE Transactions on Communications Special Issue on Content Delivery Networks*, 2003, E86-B: 1849~1858
- 11 Shen Bo, Lee Sung-Ju, Basu S. Caching Strategies in Transcoding-Enabled Proxy Systems for Streaming Media Distribution Networks. *IEEE Trans On Multimedia*, 2004, 6(2)
- 12 Vishwanath M, Chou P. An Efficient Algorithm for Hierarchical Compression of Video. In: *Proc. IEEE Intl Conf Image Processing*, 1994
- 13 Kangasharju J, Hartanto F, Reisslein M, et al. Distributing Layered Encoded Video through Caches. *IEEE Trans On Computers*, 2002, 51(6)
- 14 Rejaie R, Yu Haobo, Handley M, et al. Multimedia Proxy Caching Mechanism for Quality Adaptive Streaming Applications in the Internet. In: *Proceedings of IEEE Infocom'2000*, 2000
- 15 Podlipnig S, Boszormenyi L. Replacement Strategies for Quality Based Video Caching. In: *IEEE International Conference on Multimedia and Expo (ICME)*, 2002, 2: 49~52
- 16 Zink M, Heckmann O, Schmitt J, et al. Polishing: a technique to reduce variations in cached layer-encoded video. In: *Euromicro Conference*, 2003. 249~254
- 17 Liu J, Chu X, Xu J. Proxy Cache Management for Fine-Grained Scalable Video Streaming. In: *Proc. IEEE INFOCOM'04*, 2004
- 18 Rejaie R, Kangasharju J. On Design and Performance Evaluation of Multimedia Proxy Caching Mechanisms for Heterogeneous Networks. In: *Proceedings of IEEE International Conference on Multimedia and Expo*, 2000
- 19 Radulovic I, Frossard P, Verscheure O. Adaptive Video Streaming in Lossy Networks: versions or layers? In: *Proceeding of IEEE ICME*, 2004
- 20 Yamada T, Wakamiya N, Murata M, et al. Implementation and evaluation of video-quality adjustment for heterogeneous video multicast. In: *Proceedings of APCC*, 2002. 454~457