

CDN-P2P 混合架构下直播流媒体系统服务应急保障研究

王 群 戴秀岳

(浙江树人大学现代教育技术中心 杭州 310015)

摘 要 直播流媒体系统能够为网络用户提供丰富的直播电视节目,是发展最快的网络应用之一。然而,普遍存在的启动时延大、播放不流畅等问题,严重制约直播流媒体系统服务质量的改善。从资源分配与调度的角度,给出了直播流媒体系统资源的调度原则,提出一种基于服务器资源的应急调度方案,提高网络资源的利用率和节目的体验质量。试验结果表明,服务应急保障机制可以显著降低直播流媒体系统的启动时延,并减少丢包率。

关键词 直播流媒体系统, CDN-P2P, 服务器资源调度, 性能优化

中图法分类号 TP393 文献标识码 A

Research on Service Contingency Guarantee in CDN-P2P Hybrid Architecture Based Live Streaming

WANG Qun DAI Xiu-yue

(Modern Education Technology Center, Zhejiang Shuren University, Hangzhou 310015, China)

Abstract Live streaming provides abundant television programs for end users, and has become one of the applications which promptly develop. However, common issues, including long startup delay and unsmooth play, severely restrict the improvement of quality of service in live streaming systems. From the allocation and scheduling of resource, the principle of resource scheduling for live streaming systems is presented, and service contingency guarantee scheme for server resource is proposed, which promote the utilization of network resource and the quality of experience. The experimental results shows service contingency guarantee scheme can significantly shorten the startup delay for live streaming systems and reduce the packet loss.

Keywords Live streaming, CDN-P2P, Scheduling of server resource, Performance optimization

1 概述

近年来,利用因特网来传输电视信号已经成为一种越来越流行的网络应用,它为用户提供了内容丰富、使用便捷、交互性强的视频多媒体服务,已经形成了规模庞大的产业^[1]。传统的视频节目分发采用有线电视电缆分发,信号质量高,但系统价格高而且体系架构无法灵活应对新业务发展。随着网络技术的发展与普及,通过 IP 网络传输视频节目是一个必然的选择,但大量用户的使用将导致网络局部区域资源的短缺。由此导致了内容分发网(CDN)技术的发展。通过在网络边缘架设大量服务器,CDN 可以在保证视频的 QoS 基础上尽量消除资源热点。该方案仍具有成本较高且扩展性不够好的缺陷。在另一方面,后起的对等方到对等方(P2P)技术提供了一种廉价的视频分发解决方案,不过该方案存在无法保证视频 QoS 的缺陷。近年来,人们将希望的目光放到了 CDN 和 P2P 技术结合上,期待利用两者自身的优势来形成一种较好的视频分发系统。研究表明,这种方案的确非常有前途^[2,3],不过它仍不能满足用户对视频更高的 QoS 要求。因此,采用低成本方法实现直播视频节目数据的大规模分发,同时进一步提高直播视频的 QoS,是当前直播流媒体系统需要重点研究解决的问题。

在基于 CDN-P2P 架构的直播流媒体系统研究方面,

Bo Li 等人对在因特网上大规模运行的直播流媒体系统 New Coolstreaming^[4] 测量时发现,约有 95% 左右的数据在被播放之前能够到达用户缓存。而 Yan Huang 等人分析 PPLive 数据时发现^[5],有 20% 左右用户的数据缓存时间占用户等待和观看节目总时间的比例超过 80%,这是因为系统的启动时间过长,而且少量数据分组的缺失会迟滞直播节目播放,系统不得不等待至该分组到达后才能播放,导致用户的体验质量显著下降。因此,解决或缓解这约 20% 的用户或 5% 数据无法及时到达的问题,对于当前从根本上改善直播流媒体系统的 QoS 问题,使系统更为实用,具有重要意义。本文针对 CDN-P2P 架构下的直播流媒体系统存在的 QoS 问题进行研究,通过分析资源调度,提出了一种资源分配模型来提高其 QoS。

2 相关工作

为了减轻直播视频服务器的压力,充分利用普通对等方的资源,Zhang 等人首次将 P2P 技术引入实际的直播流媒体系统中,通过构建一个基于数据驱动的覆盖网络 DONet,很好地适应实时多媒体流数据的传播和分发^[6]。Liao 等人提出的 AnySee^[7],Nazanin 等人提出的 PRIME^[13]等,也都是利用对等方的资源来实现低成本条件下直播流媒体系统的部署和运营。但是由于 P2P 网络固有的动态性和节点扰动以及防火墙/NAT 的影响,完全依赖对等方的资源无法为用户提

王 群 硕士,讲师,主要研究方向为分布式计算。

供 QoS 保证。Skevik 等人提出了一种 CDN-P2P 混合模式的多媒体流体系结构,分发视频数据^[2]。此时,网络中部署更多的服务器资源,使 QoS 更有保证。Liu Yu 等人提出了一种两层的 P2P 辅助 CDN 体系结构,CDN 分发层位于骨干网络,而 P2P 分发层则部署在边缘的接入网上^[3]。而 Wu 等人则从不同频道之间资源的协作出发,提出了一种用户收看和数据上传相分离的方法 VUD 来实现多频道直播流媒体系统之间的资源调度,从而保证各个频道的资源供给要大于资源的需求^[4]。此外,许多研究人员从提高资源利用效率出发来改进用户收看视频的体验质量。Zhou 等人从数据调度算法角度出发,提出了一种混合策略的方法来对数据进行调度,在一定程度上提高了数据按时到达用户缓存的比例^[10]。Yang 等人则提出了一种期限感知调度机制,通过考虑数据请求的时间期限避免了资源的浪费^[11]。而 Wang 等人提出将随机网络编码技术应用到基于 P2P 的直播流媒体系统中,有效提高了系统的性能^[8]。Nguyen 等人将可扩展编码和网络编码相结合并引入到 P2P 多媒体流系统中,从而改善视频传输质量,并用试验证明了这种方法的可行性^[9]。然而上述方法都是从提高资源利用效率的角度出发改进直播流媒体系统的性能,并未考虑服务器和对等方的资源特点,也未对播放时间紧迫的数据和非紧迫的数据加以区分,导致了直播流媒体系统中仍然存在着较高的数据丢失概率,对视频 QoS 产生了较为严重的影响。

3 直播流媒体系统的资源调度原则

假设 $R(Server)$ 为系统中 CDN 服务器提供的资源, $R(P2P offered)$ 是全体端用户以 P2P 方式向系统提供的资源,而直播流媒体系统的总体资源为 $R(LS)$ 。从网络资源提供和消耗平衡的角度来看,CDN-P2P 架构下的直播流媒体系统的资源供给主要由两部分组成,一部分为 $R(Server)$,另一部分是 $R(P2P offered)$,则有

$$R(LS) = R(Server) + R(P2P offered) \quad (1)$$

而在直播流媒体系统中,所有正在收看电视节目的用户都需要消耗一定的资源来保证自身的 QoS。假定用户基于 P2P 方式流畅地观看节目所需的总体资源为 $R(P2P used)$,当不等式(2)成立时,无论如何采用何种调度策略,系统都无法满足用户 QoS 需求:

$$R(LS) < R(P2P used) \quad (2)$$

仅当满足式(3)时,采用某种调度策略,系统才有可能满足用户 QoS 需求:

$$R(LS) \geq R(P2P used) \quad (3)$$

因此,仅当在满足式(3)条件时,研究系统满足用户 QoS 需求时可能采取的资源调度措施才有意义。

由于此时要求 $R(Server) + R(P2P offered) > R(P2P used)$,但 P2P 用户的动态性、自私性等行为使得 $R(P2P offered)$ 与 $R(P2P used)$ 均在一个范围内波动,因此对于一个稳定的直播流媒体系统,进一步要满足下式

$$R(Server) + R(P2P offered)_{low} \geq R(P2P used)_{high} \quad (4)$$

其中, $R(P2P offered)_{low}$ 表示 P2P 用户能够提供资源的最小值, $R(P2P used)_{high}$ 表示 P2P 用户在视频满足 QoS 状态下使用资源的最大值。

从服务提供商的角度而言,为了取得最大的经济利益,在支持尽可能多的用户的前提下应使 $R(Server)$ 最小。由于当

$R(Server) = 0$ 时,系统无法保证 $R(P2P offered)_{low} > R(P2P used)_{high}$ 成立,因此服务提供商必须提供适量的 $R(Server)$ 才能使式(4)成立,一定量的 $R(Server)$ 是保证系统 QoS 的最后一道屏障。这表明:

(1) 利用 P2P 技术能够降低系统成本,但通过优化 P2P 用户资源调度机制,只能在一定程度上缓解直播流媒体系统 QoS 问题。

(2) 采用 CDN 技术会增加系统成本,但当 P2P 机制无法提供满足 QoS 所需资源时,所需资源应当由 CDN 服务器提供。

由此,为了以经济的方式设计高质量的直播流媒体系统,提出在 CDN-P2P 架构下直播流媒体系统资源的调度原则。

经济原则: 利用 P2P 机制,尽可能地提高 P2P 技术所提供的资源总量。

空闲原则: 尽可能保证 CDN 服务器处于轻载状态,使之可以应对用户对资源的急需。

应急原则: 当用户无法及时从对等方处获取数据时,立即向临近的 CDN 服务器发送应急请求。服务器能力应当满足应急请求的需要。

经济原则表明系统应当最大限度地利用 P2P 机制,这将能够提高 P2P 技术所提供的资源总量,从而降低系统的实现成本。空闲原则表明应当使 CDN 服务器处于低负荷状态,以便发挥服务器资源提供稳定可靠服务、对紧急数据响应快捷的特点。应急原则表明一旦出现用户需要时间上紧迫的视频数据,就应当向最近的某个服务器去请求,并以很大的概率及时获取少量的该紧迫数据,从而大大提高系统的 QoS 水平。

4 直播流媒体系统资源分配模型

4.1 三状态资源调度

在基于 CDN-P2P 架构的直播流媒体系统中,用户的状态分为初始状态、一般状态和紧急状态这 3 种情况,用户在 3 种状态下系统为用户分配资源的方式各不相同。

初始状态是指用户刚刚加入系统,其缓存中还没有充足的数据进行节目播放时的状态。此时,用户需要获得一定量的视频数据才能开始收看直播节目。根据应急原则可知,CDN 服务器需要为其分配一些资源使其尽快获得数据,以减少用户的等待时间,与此同时系统中的对等方根据经济原则也会为其提供尽可能多的资源,减轻服务器压力。

一般状态是指直播节目正常播放后,用户缓存中所缺失的视频数据距播放时间有一段较长的时间,可以及时从普通对等方处下载得到。根据经济原则和空闲原则可知,系统此时应尽可能使用 P2P 机制提供资源,并使 CDN 服务器处于轻载状态。

紧急状态是指直播节目播放过程中,缓存中的某些数据经过较长时间后仍然未能得到,因此距离播放时间很近,如果不及时获取将出现丢包的状态。此时,根据应急原则,系统启用服务器为其提供紧急资源,使缺失数据及时到达用户。

图 1 以有限状态机形式描述了直播视频系统中用户三状态资源调度和转换关系。

在系统分配资源具体过程中,视频资源主要有服务器和对等方两个来源。下面分别就这两个来源的资源调度方法展开讨论,主要从对等方资源请求策略和服务器资源分配策略进行。对等方资源请求策略是指对等方请求缺失的数据时,

根据数据的紧急程度来判断应当向服务器还是普通对等方发送请求获得资源的方法,而服务器资源分配策略是指服务器接收到各类数据请求时,合理分配自身资源以保证系统的性能和用户的 QoS 的方法。

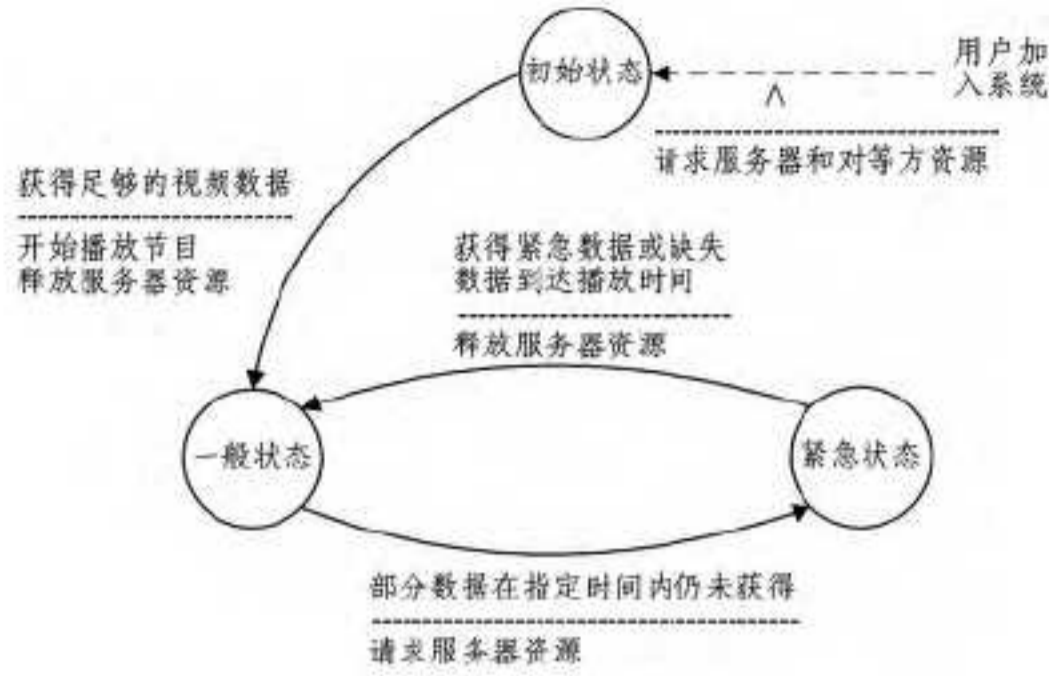


图 1 三状态资源调度和转换关系

4.2 对等方资源请求策略

对等方在刚加入系统时,如果仅向其他对等方请求数据下载会因为资源的查找和发现而花费较长的等待时间,因此需要首先与服务器建立邻居关系,并从服务器处获取最初的数据块,这类数据称为 I 型数据;随着获得更多的对等方信息,对等方优先向其他对等方请求新的数据块,服务器提供的 I 型数据的负载逐步减轻。

根据直播流媒体系统的资源调度原则可知,在一般状态下,如果对等方所需数据块的播放时间 $t_p > T_u$ (T_u 是正常播放最小保证时间),对等方就应当向具有邻居关系的对等方寻求资源,请求的数据称为非紧迫数据。在紧急情况下,如果 $t_p \leq T_u$,请求的数据称为紧迫数据,则转向服务器发出资源请求,以利用服务器能够提供稳定可靠的数据供给的特点,避免出现数据无法及时到达用户缓存的情况。来自服务器的紧迫数据被称为 II 型数据。可见, T_u 的选择极为关键, T_u 设置过大就会给服务器带来较重的负担,就要通过多设置服务器来处理服务请求,违反经济原则; T_u 设置较小则服务器负担较轻,而对等方所提供的数据增多,符合经济原则和空闲原则,但容易导致违反应急原则。图 2 显示了对等方的资源请求的情况。

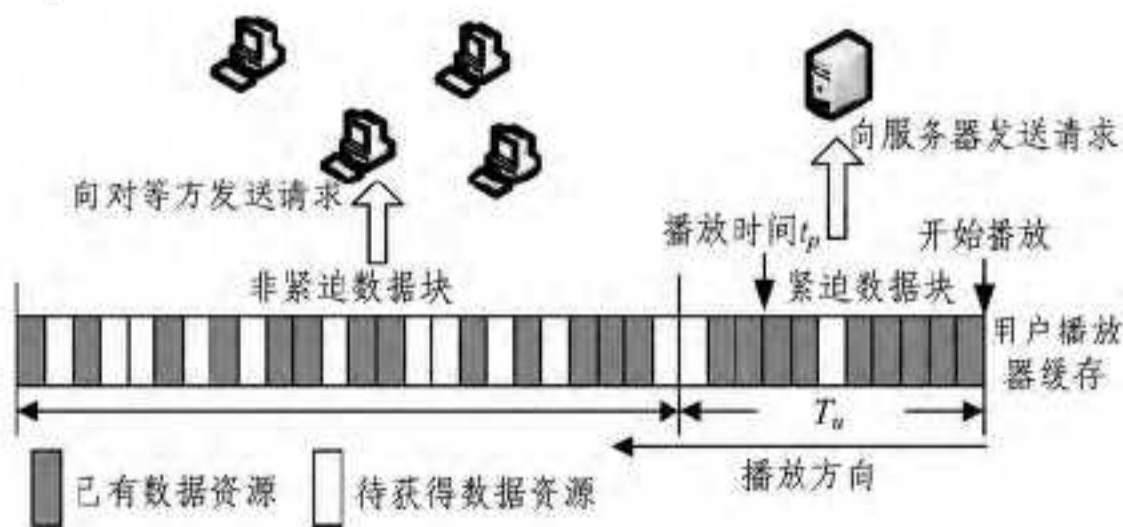


图 2 对等方资源请求策略

除上述紧迫数据外,还有部分对等方应向服务器请求系统中较为稀少的数据,称为 III 型紧迫资源,对等方在获得这些紧迫数据后可以向其他对等方转发这些数据块。如果充分利用对等方的上传能力,就能够降低这些数据随播放时间推移变成紧迫数据的概率,而上传能力越强的对等方传播数据块的速度更快,因此这些对等方应当向服务器请求稀缺数据以加速它们的转发和扩散。

4.3 服务器资源分配策略

根据直播流媒体系统资源调度原则可知,系统中服务器应当发挥的功能是当对等方出现数据紧缺时为其提供应急服

务。根据对等方资源请求策略可知,服务器需要为对等方提供 3 类数据,提供 I 型数据的目的是使对等方尽快获得开始播放所需的初始数据块,减少用户的等待时间;提供 II 型数据的目的是保证用户所需数据能够及时到达播放器缓存,避免出现数据丢失的情况;提供 III 型数据的目的是为对等方提供稀缺资源,提高数据的扩散速度,降低数据成为紧迫数据的可能性。这 3 类数据所发挥的功能不同,重要性也存在差异,因此服务器在响应这些数据请求时,应当加以区分。为了实现区分服务,采用加权公平队列的方法来对这 3 类数据请求进行分类,每一个数据请求按照所属类别加入到相应的队列当中,并为每个队列划分相应的权重,服务器根据权重值调度需要处理的数据请求。具体的实现流程如图 3 所示。

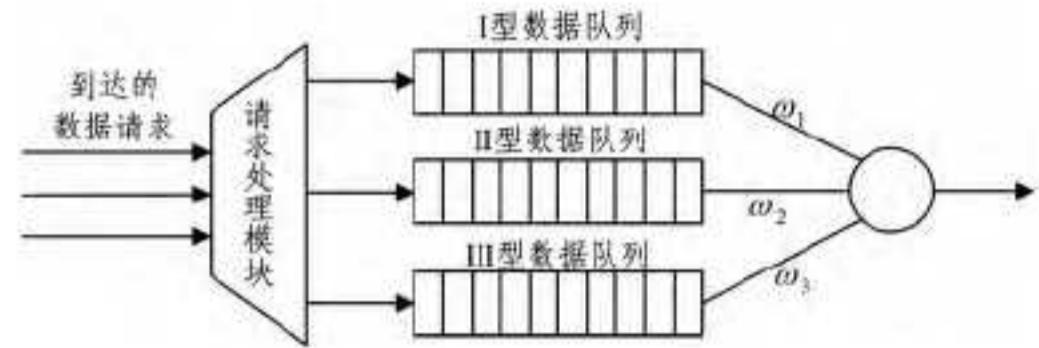


图 3 服务器的加权数据请求多队列

在采用加权公平队列的方法对两类不同的数据请求进行区分服务时需要考虑两个问题:

- 1) 各个队列应当采用何种调度算法对各自的数据请求进行排序。
- 2) 各个队列的权重应当如何分配,从而实现服务器资源的最优利用。

4.3.1 各队列的数据调度算法选择

对于 I 型数据队列来说,对等方在刚加入系统后会主动向服务器发送请求以获取 I 型数据,因此先加入系统对等方的 I 型数据请求往往会比后加入的对等方更早到达服务器,而先来先服务调度算法按照请求的到达顺序来确定下一个被服务对象。所以在 I 型数据队列中,采取先来先服务调度算法为那些等待时间更长的对等方优先提供服务。

对于 II 型数据队列来说,每个被请求的数据块距离播放时间都比较近,但是不同数据块之间的时间期限存在着差异性,部分数据块可能马上就要被播放,而另一部分数据块距离播放时间可能还有一段时间,在对这些数据请求进行响应时,如果不对其时间期限进行感知和区分,可能会导致某些数据请求不能得到及时的服务。因此在对 II 型数据队列中的数据请求进行调度时,应当首先计算每个新到达数据请求的时间期限,然后根据该期限值,将其插入到队列相应的位置中。需要注意的是,每个已经在队列中进行排队数据请求的时间期限会随着等待时间的增加而变得更近。

对于 III 型数据队列来说,队列中的所有数据都属于对等方都较为稀缺的数据,虽然在时间期限上还未达到紧迫的程度,但是将这些数据扩散给其他对等方还需要花费一定的时间,因此需要优先为序列号更小的数据提供服务,以避免这些数据由于传播不及时而变成紧迫数据。

4.3.2 各队列的权重分配

在为 3 个队列进行权重分配时,需要根据 3 类数据请求的重要程度来进行,重要性高的数据请求队列所获得的优先级也应当较高,以保证它们的 QoS,同时也要为低优先级的队列分配一定的资源来提供服务保障。通过前文对 3 类数据的分析可知,II 型数据的目的是防止对等方出现丢包的情况,而少量数据的丢失都会造成播放质量严重下降,因此 II 型数据

请求在队列中的等待时间的上限为 T_u , 否则数据到达用户播放器缓存时将错过播放时间。这里 II 型数据的等待时间被设置为 $T_u/3$, 以消除数据的传播时延以及对等方在请求这类数据不及时而带来的时间偏差, 同时如果数据在传输的过程中出现丢包, 对等方仍有足够的时间再次发送请求。III 型数据的目的是为对等方提供稀缺数据, 从而降低数据变成紧迫数据的概率, 因此服务器需要为其提供一定的资源保障, 而且 III 型数据的等待时间也不能超过 T_u , 因为如果被请求的数据恰好为距离播放时间最近的非紧迫数据, 等待时间过长也会导致数据丢失。I 型数据是为了降低用户加入系统后的播放时延, 当服务器资源充足时, 可以为其提供较多的资源以提高用户体验, 但是当服务器资源较为紧缺时, 则需要优先保障 II 型和 III 型数据的请求。为此, 需要设计一种自适应的权重分配方案, 动态地调整服务器分配给各个队列的资源数量, 从而高效地利用服务器资源。

假定服务器的资源总量为 R , 数据报文的平均大小为 P , 服务器为各个队列分配的权重大小分别为 ω_1 、 ω_2 和 ω_3 , 并且 $\sum_{i=1}^3 \omega_i = 1$, 为了保证 II 型和 III 型数据的等待时间约束, 系统定期记录这两个数据队列中数据的长度以计算各个队列的权重值, L_i' 和 L_i'' 分别表示 II 型数据队列和 III 型数据队列在第 i 次记录中报文的数量, \hat{L}_i' 和 \hat{L}_i'' 分别表示第 i 次记录的估计值。

为了保证 II 型数据队列中数据的等待时间不超过 $T_u/3$, 那么必须满足式(5):

$$\frac{\hat{L}_j' P}{\omega_2 R} \leq T_u/3 \quad (5)$$

那么 ω_2 设定的值为 $\frac{3 \hat{L}_j' P}{R T_u}$, 其中 \hat{L}_j' 为 II 型数据队列下一次记录长度的估计值, \hat{L}_j' 的值采用权重因子为 μ 的指数滑动平均来进行计算, 具体如式(6)所示。

$$\hat{L}_j' = \mu \hat{L}_{j-1}' + (1-\mu) L_{j-1}' \quad (6)$$

采用相同方法可得到 ω_3 设定的值为 $\frac{\hat{L}_j'' P}{R T_u}$, 其中 \hat{L}_j'' 为 III 型数据队列下一次记录长度的估计值, \hat{L}_j'' 的计算方法如式(7)所示。

$$\hat{L}_j'' = \mu \hat{L}_{j-1}'' + (1-\mu) L_{j-1}'' \quad (7)$$

因此, I 型数据队列所获得的权重值 $\omega_1 = 1 - \frac{3 \hat{L}_j' P}{R T_u} - \frac{\hat{L}_j'' P}{R T_u}$ 。

在实际的服务器资源分配方案中, 各个队列获得的权重值会根据 II 型和 III 型数据队列中报文的长度动态调整。

5 性能评价

5.1 仿真试验及其评价方法

为了评价服务器资源应急服务保障机制的性能, 基于文献[13]中提供的源代码实现了基于事件驱动的 P2P 多媒体流仿真器, 该仿真器能够模拟分组级的数据传输以及对等方之间的端到端时延。通过对代码进行补充和完善, 使得对等方和服务器按照前文所设定的方式进行数据请求和资源调度。

在仿真实验中, 与一般的 P2P 系统相同, 将所有对等方的上传带宽设为瓶颈链路, 同时为了将高性能对等方和普通对等方区分开, 将对等方按照上传带宽的大小进行分类, 每类对等方拥有一定的比例, 表 1 显示了各类对等方的上传带宽大小和它们在系统所占的比例。

表 1 各类对等方的上传带宽大小和在系统中的比例

上传带宽大小	系统中的比例
768kbps	0.1
384kbps	0.3
128kbps	0.6

在试验中, 总共设置了 2000 个对等方, 系统中有一个服务器进行数据的分发, 视频节目的播放码率为 300kbps。对等方之间通过定期交换缓存映射表来感知其他对等方缓存中所拥有的数据块, 并且采用拉模式来请求和获取数据。此外, 为了模拟对等方的动态性特征, 对等方随机加入和退出系统。

为了对系统的性能和用户体验质量进行评价, 选择了如下两种测度:

· 启动时延: 对等方在刚加入系统时, 缓存中没有任何数据, 它需要从服务器和其他对等方处下载足够的数据才能够开始播放, 对等方从加入系统到视频节目开始播放这一段时间差就是启动时延, 启动时延越短, 用户的体验质量越好。这里设定当对等方下载并缓存了 15 秒钟的视频数据之后才开始播放。

· 分组成功到达率: 为了保证视频节目能够流畅播放, 数据块必须在被播放之前到达对等方的缓存中, 否则即使数据块最终到达了对等方处, 对于对等方来说也属于无效的数据, 相当于数据出现了丢失, 造成播放迟滞, 影响用户的正常收看。成功到达数据比例的定义是按时到达对等方缓存的数据块数量与对等方应当接收到的数据块的比值。丢包率越小, 系统性能越好。

5.2 试验结果分析

5.2.1 服务器应急保障服务的性能

为了分析服务器应急保障机制对系统性能的影响, 将其与未采用应急保障机制的方法进行了比较。图 4 和图 5 分别显示了在服务器带宽资源不断变化的情况下, 两种方法的启动时延和分组成功到达率的累积概率分布图, 其中紧迫数据缓存的大小设置为距离播放时间小于 3 秒钟, 紧迫数据队列和高性能对等方队列的权重比值为 3:7。

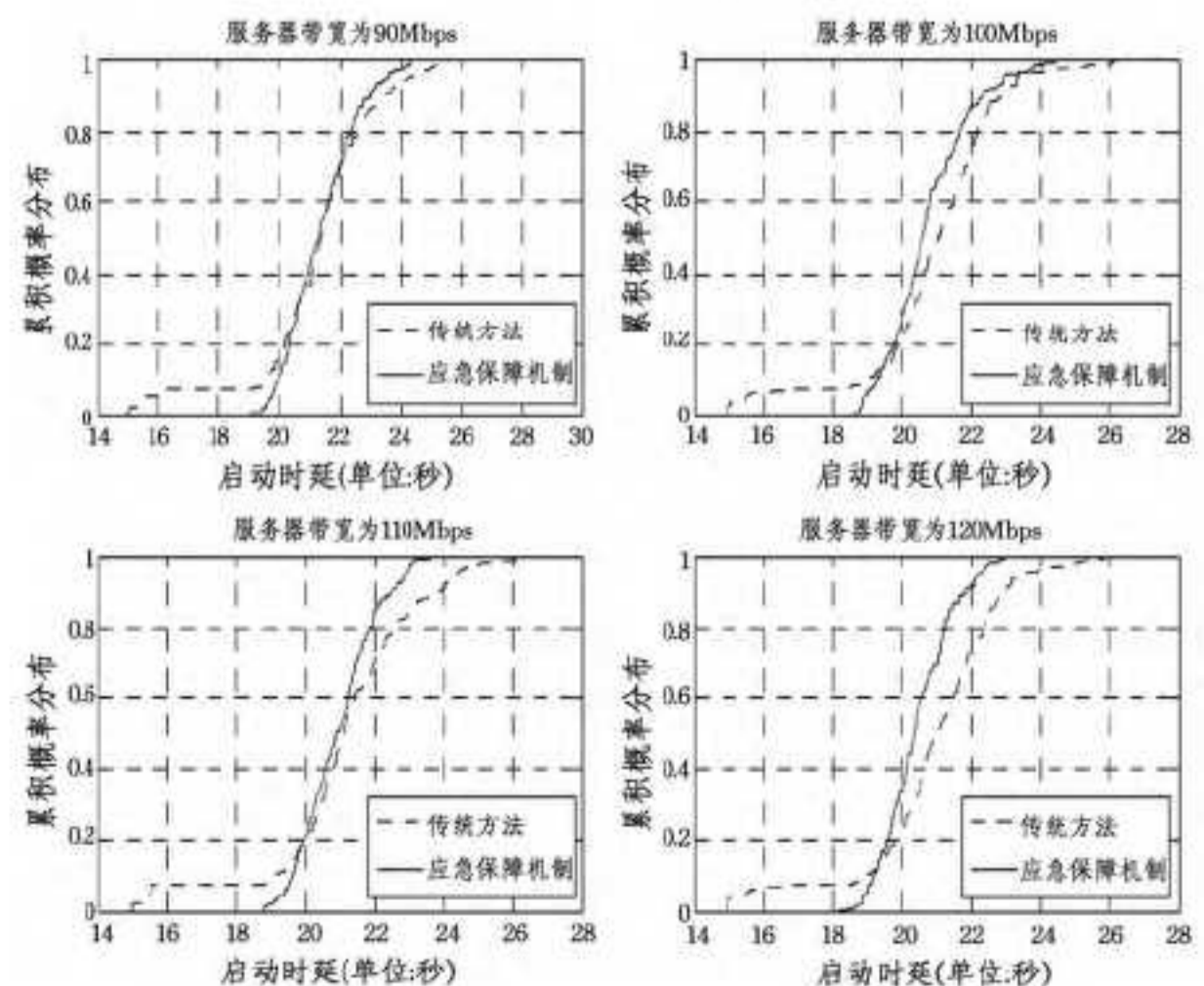


图 4 启动时延累积概率分布图

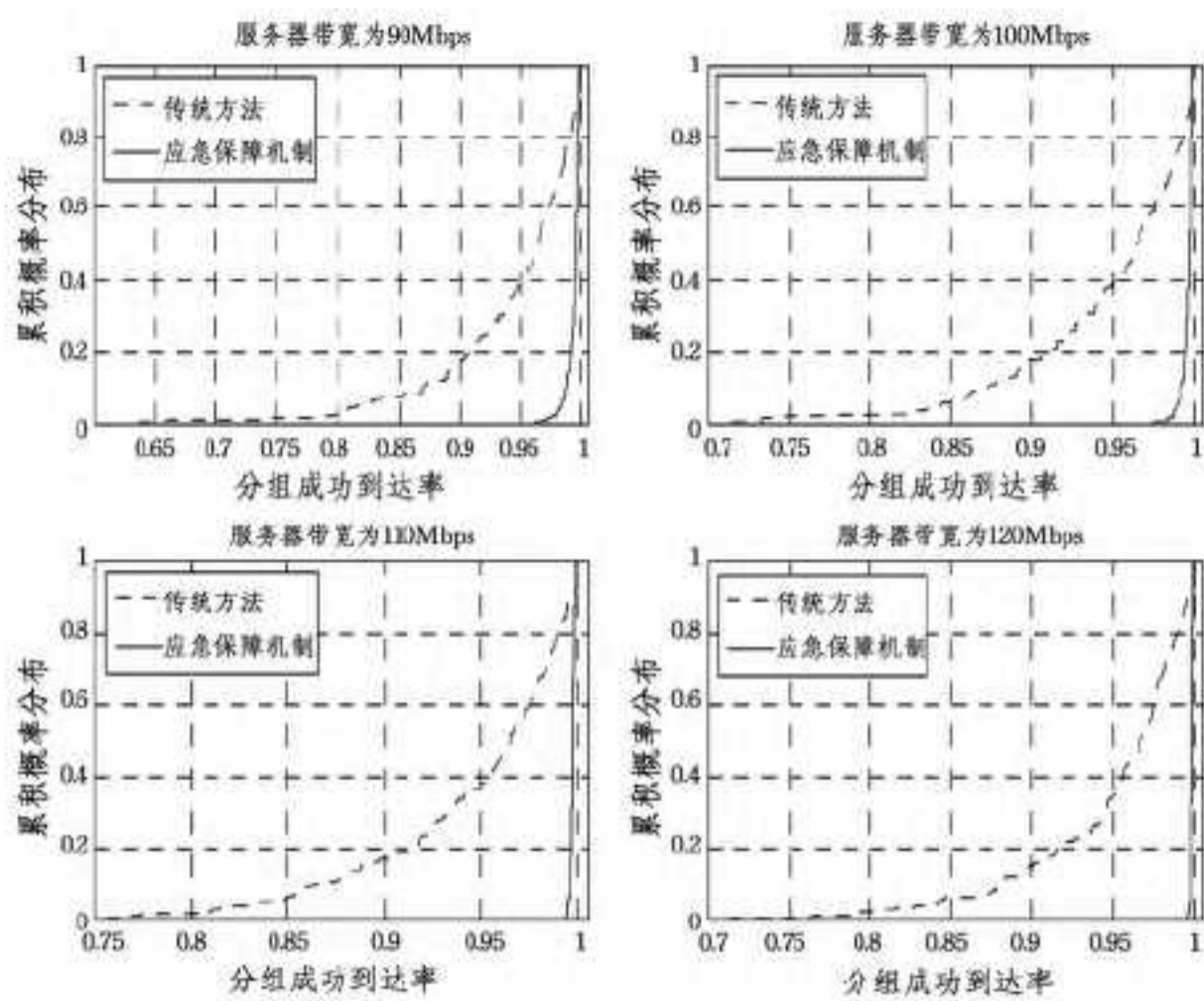


图5 分组成功到达率累积概率分布图

从图4可以看出,采用传统方法的启动时延变化范围较大,启动时延的分布从15秒至25秒不等,而且随着服务器带宽资源的增加,启动时延几乎得不到改进,说明服务器的带宽资源没有被充分利用起来。而采用服务器应急保障机制的直播流媒体系统在均值上具有更短的启动时延,对等方之间的启动时延相差也不大,而且服务器带宽资源的不断增加会显著地改进系统的性能,降低对等方的启动时延,这是因为采用服务器应急保障机制更好地利用了服务器和对等方的资源,提高了用户的QoS。

从图5可以看出,前文所提出的系统设计原则通过利用服务器性能强的特点为用户缺失的关键数据提供了充足的资源,降低了用户数据丢失的现象,极大提高了分组的成功到达率,几乎所有的用户数据丢包率都保持在2%以内,而且服务器带宽资源的增加同样可以有效促进系统性能的改进。

5.2.2 紧迫数据缓存大小设置的影响

通过前文的分析可知,紧迫数据缓存大小的设置会直接影响用户缓存中哪些数据应当从服务器处获得,哪些数据应该向对等方请求。如果紧迫数据缓存设置得较大,数据在非紧迫缓存中逗留的时间就会相对较短,它们在这一阶段得到对等方服务的可能性就会降低,紧迫数据缓存中缺失的数据也就会相对增加,这些数据都将由用户向服务器发送请求获得,如果所有对等方缺失的数据过多,那么可能会造成服务器资源的瞬时不足,并最终造成数据的缺失。而如果紧迫数据缓存设置得过小,那么尽管数据通过P2P方式获得的可能性会有所提高,但是数据允许的等待时间也就相应减少了,一旦缺失的数据进入紧迫数据缓存,必须在短时间内从服务器获得,否则同样会出现丢包的情况。因此,紧迫数据缓存大小对于服务器资源的利用率具有重要影响。通过调整缓存的大小,以得到启动时延和分组成功到达率的变化情况,来分析紧迫数据缓存大小对系统性能的影响。图6显示了启动时延和分组成功到达率的累积概率分布,其中服务器的上传带宽为100Mbps,紧迫数据队列和高性能对等方队列权重比为3:7。

从图6可以看出,紧迫数据缓存大小的设置对于系统的性能具有非常明显的影响,当紧迫数据缓存设置从7秒钟变为3秒钟时,系统的启动时延和分组成功到达率会得到显著的提高,而当数据缓存大小变为1秒钟时,系统的性能又会逐

渐降低。

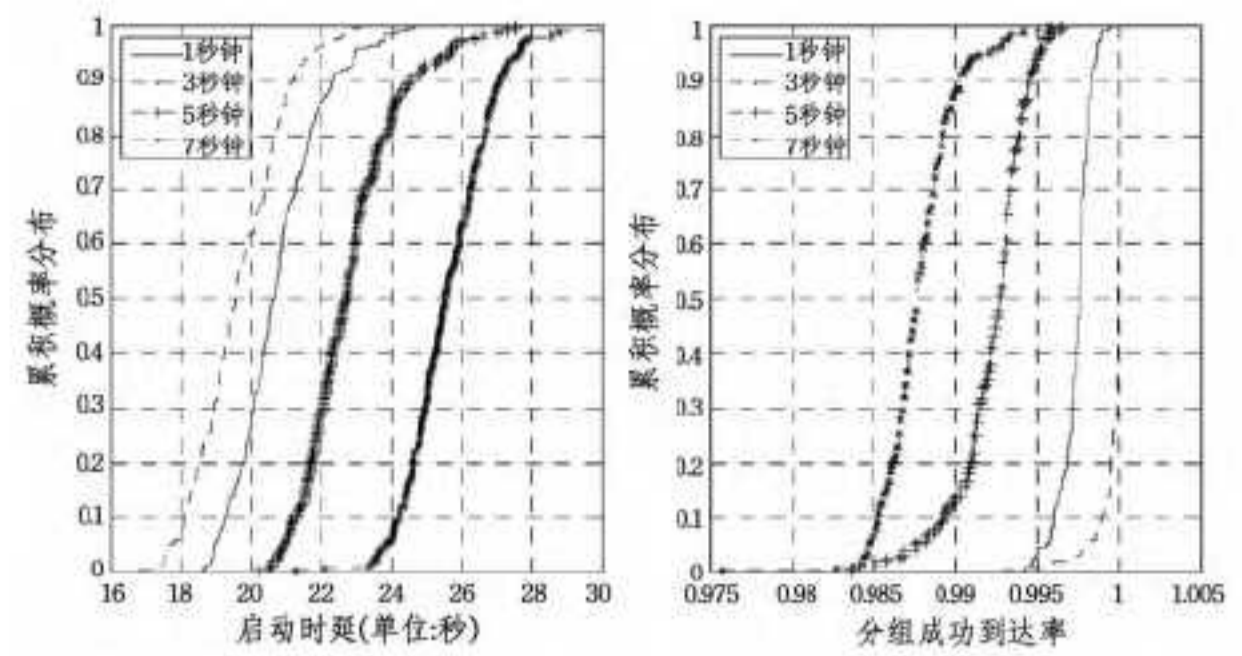


图6 紧迫数据缓存大小的影响

5.2.3 各个队列权重值设定的影响

服务器为两类队列分配不同的权重值会对系统带来一定程度的影响,当服务器的资源划分合理时,不仅可以充分利用高性能对等方的带宽优势,还可以有效避免紧迫数据由于无法得到及时响应而出现丢包的情况。为了分析紧迫数据队列和高性能对等方队列的权重比值变化对系统性能的影响,对权重比不同情况下系统的启动时延和分组成功到达率进行了分析。图7显示了对队列权重进行调整后系统的性能状况,其中服务器的上传带宽设定为100Mbps,紧迫数据缓存的大小设置为3秒钟。

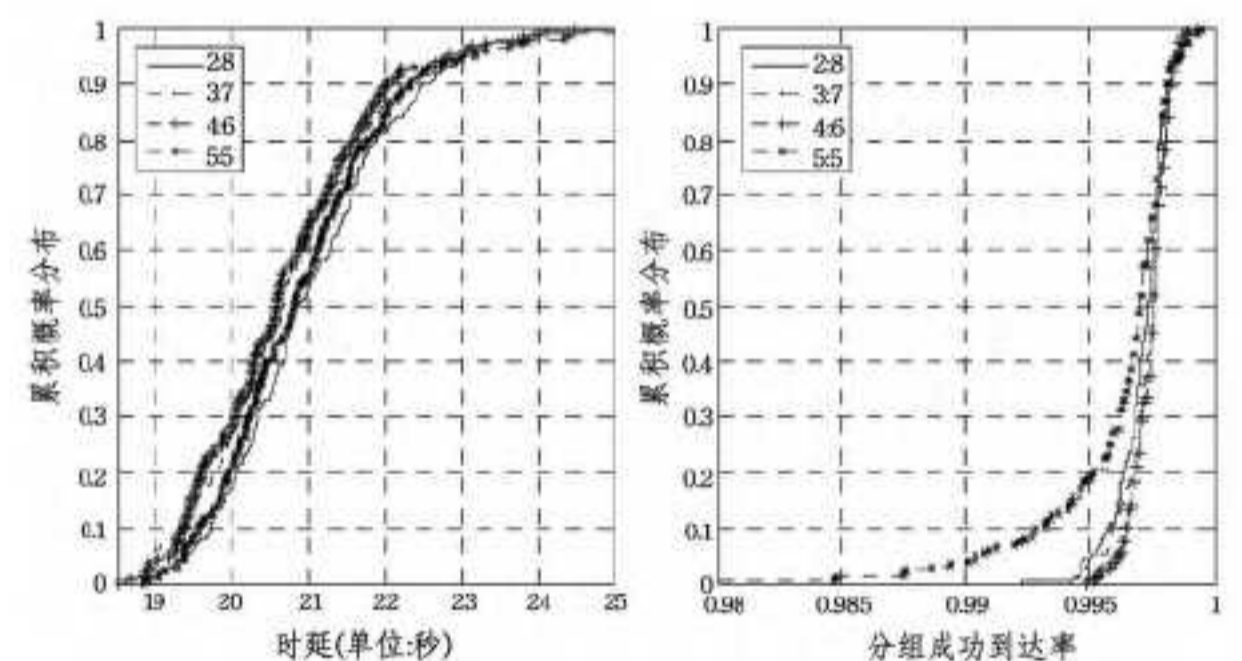


图7 权重值的影响

从图7可以看出当紧迫数据队列和高性能对等方队列从服务器分配到的带宽比为4.6时,启动时延和分组成功到达率均能够得到最好的结果,权重值的增加和减少都会降低系统的性能,这说明在这种带宽分配方案下,能够达到服务器资源利用的最优。

结束语 为了有效降低直播流媒体系统的运营成本,CDN-P2P模式成为了直播内容服务提供商首选的实现方式,但是当前的直播流媒体系统仍然由于少数丢包情况造成用户体验质量受到极大影响,其中的原因在于缺乏相应的机制来对CDN-P2P模式下服务器和对等方资源进行合理的调度和使用。针对这一情况,本文从资源的供给和消耗角度出发,对直播流媒体系统的设计原则进行了总结归纳,并提出了一种服务器应急保障机制,通过对服务器和对等方资源进行合理的分配,极大地提高了系统中分组成功到达率,同时减少了用户收看直播节目所需的等待时间,有效提高了系统的性能,改进了用户的QoS。

参考文献

- [1] Multimedia Research Group Inc. [OL]. <http://www.mrgco.com/iptv/gf1210.html>

- [2] Skevik K A, Goebel V, Plagemann T. Design of a Hybrid CDN [C] // 2nd International Workshop on Multimedia Interactive Protocols and Systems. Grenoble, France, 2004; 206-217
- [3] Liu Yu, Yin Hao, Zhu Guang-xi, et al. Peer-assisted content delivery network for live streaming: architecture and practice [C] // International Conference on Networking, Architecture and Storage. Chongqing, China, 2008; 149-150
- [4] Li Bo, Xie Su-su, Yang Qu, et al. Inside the New Coolstreaming: Principles, Measurements and Performance Implications [C] // Proc. of IEEE INFOCOM. Phoenix, AZ, USA, 2008; 1031-1039
- [5] Huang Yan, Fu T Z J, Chiu D-M, et al. Challenges, Design and Analysis of a Large-scale P2P-VoD System [C] // Proc. of ACM SIGCOMM. Seattle, Washington, USA, 2008; 375-388
- [6] Zhang Xin-yan, Liu Jiang-chuan, Li Bo, et al. CoolStreaming/DONet: A Data-Driven Overlay Network for Efficient Live Media Streaming [C] // Proc. of IEEE INFOCOM. Miami, USA, 2005; 2102-2111
- [7] Liao Xiao-fei, Jin Hai, Liu Yun-hao, et al. AnySee: Peer-to-Peer Live Streaming [C] // Proc. of IEEE INFOCOM. Barcelona, Spain, 2006; 1-10
- [8] Wang M, Li Bao-chun. R²: Random Push with Random Network Coding in Live Peer-to-Peer Streaming [J]. IEEE Journal on Selected Areas in Communications, 2007, 25(9): 1655-1666
- [9] Nguyen A T, Li Bao-chun, Eliassen F. Chameleon: Adaptive Peer-to-Peer Streaming with Network Coding [C] // Proc. of IEEE INFOCOM. San Diego, CA, USA, 2010; 1-9
- [10] Zhou Yi-peng, Chiu D-M, Lui J C S. A Simple Model for Chunk-Scheduling Strategies in P2P Streaming [J]. IEEE/ACM Transactions on Networking, 2011, 19(1): 42-54
- [11] Yang Yan, Chow A L H, Golubchik L, et al. Improving QoS in BitTorrent-like VoD Systems [C] // Proc. of IEEE INFOCOM. San Diego, CA, USA, 2010; 2061-2069
- [12] Peer-to-Peer Streaming Simulator [OL]. <http://media.cs.tsinghua.edu.cn/~zhangm/download/>
- [13] Magharei N, Rejaie R. PRIME: Peer-to-Peer Receiver-Driven Mesh-Based Streaming [J]. IEEE/ACM Transactions on networking, 2009, 17(4): 1052-1065
- [14] Wu Di, Liang Chao, Liu Yong, et al. View-Upload Decoupling: A Redesign of Multi-Channel P2P Video Systems [C] // Proc. of IEEE INFOCOM. Rio de Janeiro, Brazil, 2009; 2726-2730

(上接第 458 页)

引入本体思想, 将本体植入关系数据库, 为其增加语义结构, 并且在该语义结构的基础之上, 将其映射到描述逻辑上, 利用现有的逻辑推理系统进行推理, 为关系数据库增加了语义功能, 使其支持自动推理。

但本文同时也存在不足之处, 即在语义数据层中, 对于复合对象属性无法在 E-R 图中有效地表示出来。例如, 若满足表达式 $\text{has-child}(x, y) \wedge \text{has-parent}(x, z)$ 则可以推出表达式 $\text{has-grandparent}(y, z)$, 这样的复合对象属性推理不能在语义层中很好地表示; 另外, 在定义结构数据和实例数据时, 我们假定的是由专家有效地操作, 这个在实际情况下还是会存在问题。在领域内的规范化定义和关系数据的自动化抽取方面是有待提高的。

虽然存在不足, 但是可以改进并展望。在本文实验中, 通过有效的定义类、规则等, 可以达到统计一个人的亲戚关系网, 那么不止亲戚关系, 我们可以将其扩展到同学、同事等各种人际关系, 这样就有望形成一个人的人际关系网, 随着人员信息的增加、推理、积累, 人际网关系越发完善, 这是一件非常有意义的事。

参 考 文 献

- [1] Codd E F. A relational model of data for large shared data banks [J]. Communications of the ACM, 1970, 13(6): 377-387
- [2] Codd E F. Further normalization of the database relational model[R]. IBM Research Report. San Jose, California, 1971
- [3] Codd E F. Extending the database relational model to capture more meaning[J]. ACM Trans. Database Syst., 1979, 4(4): 397-434
- [4] 孔繁胜. 知识库系统原理[M]. 杭州: 浙江大学出版社, 2000
- [5] Berners-Lee T. Weaving the Web[M]. Harpur, San Francisco, 1999
- [6] Berners-Lee T, Hendler J, Lassila O. The Semantic Web[M] // Scientific American. May 2001
- [7] Antoniou G, van Harmelen F. A Semantic Web Primer[M]. Beijing: China Machine Press, 2008
- [8] 史一民, 李冠宇, 刘宁. 语义网服务中本体服务综述[J]. 计算机工程与设计, 2008, 29(23): 5976-5980
- [9] Uschold M, Gruninger M. Ontologies: Principles, method, and applications[J]. Knowledge Engineering Review, 1996, 11(2): 93-155
- [10] N Guarino. Formal ontology and information systems[C] // Proc of the 1st Int'l Conf on Formal Ontology in Information Systems. Trento, Italy; IOS Press, 1998; 3-15
- [11] 李善平, 胡玉杰, 郭鸣, 等. 本体论研究综述[J]. 计算机研究与发展, 2004, 41(7): 1041-1052
- [12] Neches R, Fikes R E, Finin T, et al. Enabling Technology for Knowledge Sharing[J]. AI Magazine, 1991, 12(3): 36-56
- [13] Gruber T R. A Translation Approach to Portable Ontology Specifications[J]. Knowledge Acquisition, 1993, 5: 199-220
- [14] 史忠植, 董明楷, 等. 语义 Web 的逻辑基础[J]. 中国科学(E 辑), 2004, 34(10): 1123-1138
- [15] Baader F, Nutt W. Basic Description Logics[M] // Baader F, Calvanese D, McGuinness D, et al., eds. The Description Logic Handbook, Chapter 2. Cambridge Univ Press, 2003
- [16] 石莲, 孙吉贵. 描述逻辑综述[J]. 计算机科学, 2006, 33(1): 194-197
- [17] Horrocks I, Sattler U, Tobies S. Reasoning with individuals for the description logic shiq[C] // David MacAllester, eds. Proceedings of the 17th International Conference on Automated Deduction (CADE-17), number 1831 in Lecture Notes in Computer Science. Germany: Springer Verlag, 2000; 482-496