基于P2P网络的视频点播系统设计

万川龙,桑 军,向 宏,胡海波
(重庆大学 软件学院,重庆 400044)

摘 要:视频点播(VOD)是以用户需求为主导的视频系统,如何提高视频点播(VOD)系统的可扩展性和在动态环境中的可靠性,成为视频点播系统大规模应用的关键。文中提出了一种新的基于P2P(Peer to peer)的VOD(Video-on-demand)系统,并阐述了系统设计所采用的相关技术与方法。系统中考虑了节点均衡负载对系统整体性能的影响,采用服务器集中调度与节点分布协调管理相结合的资源定位方式,灵活的候选父节点策略使节点失效后能进行快速的失效恢复,提高了VOD系统的可扩展性和动态环境中的可靠性。

关键词:P2P;视频点播;均衡负载;资源定位;失效恢复
中图分类号:TP393 **文献标识码**:A **文章编号**:1673-629X(2009)06-0017-05

A P2P Based Video-on-Demand System

WAN Chuan-long, SANG Jun, XIANG Hong, HU Hai-bo
(School of Software Engineering, Chongqing University, Chongqing 400044, China)

Abstract: Video-on-demand (VOD) is a user-demand driven video system. The key to the large-scale application of VOD system is dependent on how to improve the scalability and reliability in the dynamic environment. A new P2P(peer-to-peer)-based VOD(video-on-demand) system is presented in this paper. It details related technology and method used in the system designing and realization. In this system, the influence of node balanced load on overall performance was taken into consideration, resources location mode of combining server scheduling with coordination management for node distribution was adopted, and the flexible strategy for the candidate father nodes ensures a quick failure recovery after the node invalidation. It improved the scalability and reliability in the dynamic environment.

Key words: P2P; viedo-on-demand; balanced load; resources location; failure recovery

0 引 言

视频点播(Vedio-on-demand, VOD)是一种基于流媒体技术而实现的网络多媒体应用。传统的视频点播系统基于C/S模式,它为每个请求分配一条数据流,当用户大规模增长,特别是点播流行节目时,系统将难以提供质量保证,因而无法满足大量的点播服务。

组播为VOD大规模应用提供了可能,组播技术可分为IP网络层组播和应用层组播[1]。基于IP组播的VOD系统在网络层建立组播树,以多路复用的方式共享媒体流,能大量节约带宽和减轻服务器负载,如PATCHING[2]。但由于路由器支持不足、拥塞控制、可靠性管理等问题,基于IP组播的VOD服务难以在短期内广泛实用。P2P流媒体[3]通过利用普通节点(PEER节点)的闲散资源(带宽、CPU以及存储资源等)为其他的节点提供服务,有效地减小了服务器的负载。

但是基于P2P的VOD系统也面临着很多的问题:

1)如何在P2P这样一个动态的环境中进行内容管理以及资源定位;

2)在节点离开或失效的情况下如何进行其关联节点的数据恢复以保证其服务质量;

3)如何在网络异构的环境中实现均衡负载。

针对VOD点播中异步的用户请求,人们提出了多种解决方案,其核心是让PEER节点缓存部分流媒体数据[4],从而为其它节点提供数据服务。其中P2CAST[5]是一种基于PATCHING流的P2PVOD数据分发体系。P2VOD[6]是一种PEER节点FIFO队列

收稿日期:2008-10-04;**修回日期**:2009-03-20
基金项目:国家高技术研究发展计划(863计划)资助项目(2007AA01Z445)
作者简介:万川龙(1980-),男,硕士研究生,研究方向为数据通信与网络、信息安全;桑 军,副教授,博士,研究方向为信息安全、软件工程、图像处理;向 宏,教授,博士,研究方向为信息安全、系统工程、软件工程。

不等长的 VOD 服务体系。DIRECTSTREAM[7]是一种基于中心索引的 VOD 服务体系，服务器记录了所有 PEER 节点上数据缓存的信息。P2PSTREAM[8]由中心服务器统一分配节点应该缓存哪些数据。PEERVOD[9]对 P2VOD 进行了改进，考虑了节点失效后数据恢复的完整性。但是 PEERVOD 中簇首节点根据节点请求计算出候选的父节点，降低了资源定位的准确率，同时增大了簇首节点成为系统的瓶颈的概率。

针对上述问题，提出了一种新的基于 P2P 的 VOD 系统，它具有如下几个特点：

1)考虑了节点带宽对系统整体性能的影响；

2)在资源定位上采取中心索引与分布管理调度相结合，与 PEERVOD 方式不同；

3)在节点加入系统时，初步的资源定位由服务器完成，之后由簇首节点以及普通节点协同管理调度；

4)采用动态的候选父节点策略，在节点离开或失效的情况下能快速进行失效恢复；

5)采用了单源单路径与多路径[10]相结合的内容分发策略。

1 系统模型

系统由服务器(S)与用户节点(PEER)构成，其中用户节点根据管理控制功能分为簇首节点以及普通用户节点。

图 1 为系统模型图，为了便于系统描述，给出系统的几个定义：

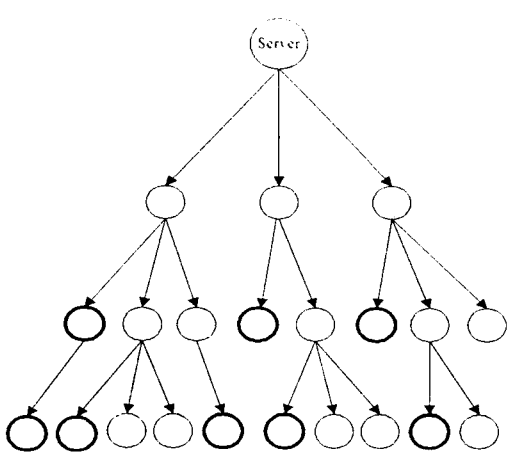


图 1 系统模型图

定义 1 系统中直接从服务器 S 一个通道获取数据的 PEER 节点集合称为一个簇，簇中直接从服务器 S 获取数据的节点称为簇首节点 HEAD_CLUSTER。

定义 2 对于系统中任意节点 P_i 和 P_j，如果在当前时刻满足 $MIN(P_j) \leqslant NEXT(P_i) \leqslant MAX(P_j)$，且 P_j 不是 P_i 的父节点，则称 P_i 是 P_j 的候选父节点。

1.1 模型定义

在 P2PVOD 系统中首先需要解决的问题是让新加入的 PEER 节点从哪个或哪些节点获取节目数据，以及允许哪些节点提供数据服务。特别是对于动态变化的 P2PVOD 系统，要求实时性更高、节点协调能力更强，因此要求高效的协议来管理节点和实现节点的资源定位。

设计了中心索引和分布式相结合的控制协议，具体描述如下：

1)服务器 S。接收或转发节点请求，维护直接与服务器相连的节点的状态信息。服务器维护的必要信息包括：簇 ID、簇首节点的 IP 地址、簇所缓存的最小数据块的编号以及最大数据块编号。通过维护簇缓存的数据块的范围，可以有效减小资源定位搜索范围以及不必要的网络资源浪费。

2)簇首节点 HEAD_CLUSTER。与服务器交换簇状态信息、请求流数据、接收或转发服务器转发的请求；维护本簇中的连续缓存的最小数据块、最大数据块编号、子节点的 IP。

3)普通用户节点。维护父节点以及候选父节点 IP、子节点 IP、子节点缓存的数据范围。从系统返回的节点集选择父节点并获取数据，并将该节点集作为候选父节点的信息进行维护，以便在父节点离开或失效时进行快速失效恢复。当父节点传输数据速度持续小于某一特定值时，选择一个候选父节点并进行数据请求。

4)节点间通信。节点与父节点、候选父节点、子节点维持心跳信息，以确定对方是否在系统之中，当在设定的时间间隔内没有收到对方的心跳信息，则认为该节点已失效，将其从自己维护的数据中删除。心跳信息内容包括：节点缓存数据块范围、剩余带宽、下一个数据块编号。

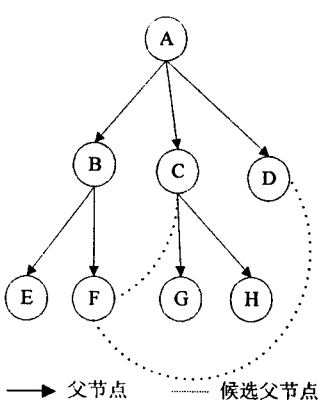


图 2 节点 F 有一个父节点 B 和两个候选父节点 C、D

图 2 描述了节点 F 有一个父节点以及多个候选父节点的情形，F 同时与父节点 B 以及候选父节点 C、D

维持通信。

5)节点均衡负载。

系统根据节点的网络带宽评定加入请求优先级,在服务器以及资源定位树中,将考虑节点请求优先级,减小优先级低的节点直接加入服务器的概率。系统中优先级由低到高分为 0、1、2 级,假设 P_i 的加入请求优先级为 0,当不存在簇满足该节点请求时,服务器将延迟处理该请求;当存在簇满足该请求时,向下转发并将优先级加 1,当优先级大于 1 时,将按照最小树高原则处理该请求;通过节点优先级机制,可以减小优先级低的节点成为组播树瓶颈的概率。系统通过服务器的初始资源定位并转发给由簇首节点、附属簇首节点组成的资源定位树,减小了节点因遍历整个组播树而造成网络资源的浪费,提高了资源定位效率;通过优先级机制实现了节点的均衡负载,有效地提高了系统的可扩展性和稳定性。

1.2　节点加入算法

新节点在加入系统前,首先试图获取多个候选父节点的信息,在节点发出请求的同时启动一个计时器,在规定的时间范围内接收并处理系统返回的信息。设服务器 S 接收到节点 P_i 的加入请求,具体算法描述如下:

1)服务器首先检查用户请求类型,如果是 REJOIN_SYSTEM,设置请求 TTL 值,转 2);如果是 REJION_SERVER 则查看服务器是否有空闲的资源,若有则直接向 P_i 返回加入成功;否则转 8);

2)服务器查看簇首节点信息表,满足条件:MIN(CLUSTER)≤REQUEST(P_i)≤MAX(CLUSTER),则转 3);否则转 7);

3)向该簇首节点转发用户请求信息;簇首节点查看自己是否满足请求条件:STATE_HEAD_CLUSTER! =FALSE&&MIN(HEAD_CLUSTER)≤REQUEST(P_i)≤MAX(HEAD_CLUSTER),若满足,则返回加入成功;否则转 4);

4)选择合适的子节点并向下转发,若存在节点 Pj 满足条件:TTL > 0 && STATE_PEER ! =FALSE && MIN(P_j)≤REQUEST(P_i)≤MAX(P_j),则转 5);否则 TTL=TTL-1,若 TTL≤0 则转 7);否则,重复 4);

5)将该用户请求转发给 P_j,P_j 返回加入成功的信息,转 6);

6)请求节点在限定的时间内收集到节点集合 Ω,若 Ω 为空则转 7);若 Ω 非空,则按照一定的原则选择父节点 FATHER,并将从集合 Ω - FATHER 选择部分节点作为自己的候选父节点,算法结束;

7)节点改变请求类型为 REJOIN_SERVER,转 1);

8)节点加入失败;

在服务器转发节点请求时,为其设置 TTL 值,可以避免大量的节点集中聚集在一棵组播子树上,同时通过 TTL 值可以有效控制树的高度。TTL 值的设置过大,则组播树高度值也相应增大;TTL 值过小,则直接与服务器连接的节点会增多,从而增加服务器的压力,因此 TTL 值的设定应根据系统的当前情况做出判断。

1.3　节点失效恢复

节点失效恢复是所有 P2PVOD 系统都需要重点考虑的问题,它涉及到系统的容错性能以及稳定性。节点失效可分为用户主动退出、非正常退出、节点播放结束三种情形,下面将分别讨论:

1) 用户 P_i 主动退出。

由于用户 P_i 主动退出,节点在退出前将通知父节点 P_f、所有的子节点 P_s,$P_s \in SON(P_i)$ 以及该附属簇中其它节点。P_f 在收到信息后,将该节点从子节点信息表中删除,$SON(P_f) = SON(P_f) - P_i$;子节点在接收到该信息后,检查根据候选父节点集 CANDIDATE(P_s),若非空则发送数据请求信息,并根据返回信息选择父节点,如果所有的候选父节点都连接不成功,则直接向服务器进行数据请求。

2) 用户因为非正常原因退出。

由于用户因为非正常原因退出系统,而无法通知相关节点。以节点 P_i 失效为例进行说明。由于 P_i 与其父节点 P_f、子节点 P_s,$P_s \in SON(P_i)$ 定时地发送心跳数据信息,因此在设定的时间间隔内若没有收到心跳信息,则认为 P_i 失效。P_f 将该节点从子节点信息表中删除,$SON(P_f) = SON(P_f) - P_i$;子节点将检查侯选父节点集 CANDIDATE($P_s$),若 CANDIDATE($P_s$) 非空则发送数据请求信息,并根据返回信息选择父节点;如果候选父节点为空,则直接向服务器进行数据请求。

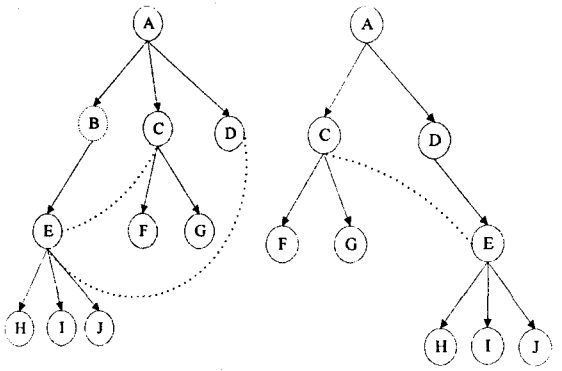
3) 节目播放结束引起节点退出由于组播树从上至下的流分发方式,因此对于系统中任意节点 P_i 和 P_j,如果 P_i 为 P_j 的上游节点,则有 $MIN(P_i) > MIN(P_j)$ 且 $MAX(P_i) > MAX(P_j)$,因此总是上游节点首先点播完而退出。当节点 P_i 节目播放结束时之前,其子节点必须寻找新的父亲节点。但这是一种可以预知的情形,下面提供了两种应对策略:

a) 当 T - NEXT(P_i) 小于某一系统设定值时,节点 P_i 将在子节点集合 SON(P_i) 中计算出合适的继承节点 P_j。

b) 节点 P_i 提前告知其子节点 P_j,$P_j \in SON(P_i)$,

P_j 检查 CANDIDATE(P_j),若 CANDIDATE(P_j) 非空,则发送数据请求,并根据返回信息选择父节点;若候选父节点为空,则直接向服务器进行数据请求。

图 3 描述了节点 B 失效后,其子节点 E 加入系统的状态,当节点 E 的候选父节点数量小于设定值时,E 将向服务器发出请求信息以获取足够的候选父节点,保证其父节点再次失效后能成功加入系统。



(a)节点B失效 (b) 节点E加入系统

图 3 节点 B 失效后,其子节点 E 加入系统的状态图

可以看到,当节点 B 退出系统后,只有其子节点 E 进行了调整,而 E 的子节点随着 E 的调整,能继续通过 E 获取数据。避免了大规模节点失效而给系统带来不稳定的情况。

2 系统架构设计

系统设计的思想是基于上述提出的系统模型,目的是构建一个基于 P2P 网络模型的流媒体点播系统,它满足 3 个条件:

(1)有效的:能保证节点快速地加入和具有较好的数据传输延迟性能。

(2)健壮的:保证绝大多数节点在动态环境下的流媒体服务质量。

(3)可扩展:随着系统中节点数目的增多,系统的性能保持相对稳定。

在此基础上,对系统模型进行了分层,图 4 描述了基于分层的系统构架图,系统是构建于 TCP/UDP 协议层之上,并由下至上分为 3 层:P2P 节点管理控制层、P2P 数据流管理层、媒体应用层。

* P2P 节点管理控制层:负责构建基于 P2P 的覆盖层,完成流媒体数据的传输和控制消息的传递。并提供索引以及资源定位服务。

* P2P 数据流管理层:负责维护节点与节点之间的逻辑关系,同时对各种媒体格式提供接口,负责流媒体数据的监控调度和缓存管理以为媒体应用层提供数据流服务。

* 媒体应用层:包括流媒体数据的输入和输出以及流媒体数据的编码、解码功能。

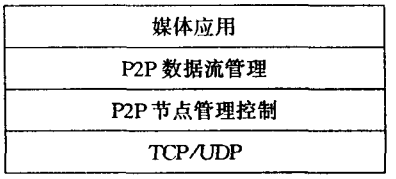


图 4 基于分层的系统架构图

系统处理流程分为服务器端与普通节点处理流程。图 5 描述了服务器端处理用户请求的流程。一个用户请求到达后,首先需要判断请求类型,如果是新的加入系统请求 REJOIN_SYSTEM,则发送满足请求条件的节点列表;如果是加入服务器请求 REJOIN_SERVER,则查看服务器是否有足够带宽容纳该节点;如果由服务器直接为该节点提供流服务,则服务器更新索引列表,保证列表的可用性。

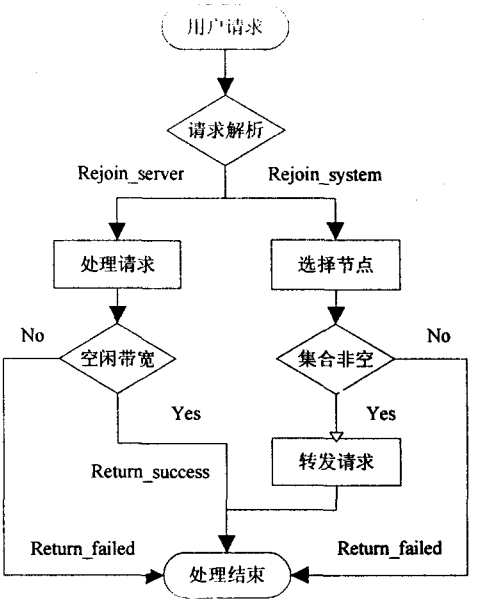


图 5 服务器端处理流程

图 6 描述了普通节点处理用户请求的流程。当用户请求到达后,首先检查自己是否满足请求条件以及是否有足够带宽,若满足则直接返回加入成功信息。若没有足够带宽,则查找子节点集合,找出满足条件的节点,并进行转发。

3 系统性能分析

通常,流媒体网络应用的性能体现在两个方面:能尽快地完成并发任务;有效地处理大量的等待(由 TCP/IP 引起,或者等待通信的另一方发下一条命令)要实现一个高性能的网络服务器,必须要选择一个能够有效地达到以上标准的构架。

系统中采用了基于线程池的并发处理技术,它以

一个预选创建的线程池处理连接请求。其中一个线程叫“接受线程”,它等待新的连接并接受它们。然后,接受线程将这个连接排队,并通知工作线程(工作线程有多个)有新完成的连接。于是,其中一个工作进程将一个连接从队列中摘下,接着在此连接读取请求并处理,处理完后再接受下一个请求(如这个连接是持续连接的话)或者再从队列中取下新的连接。线程池技术能有效地提高系统节点(特别是服务器)的并发处理能力,减少处理连接的延时。

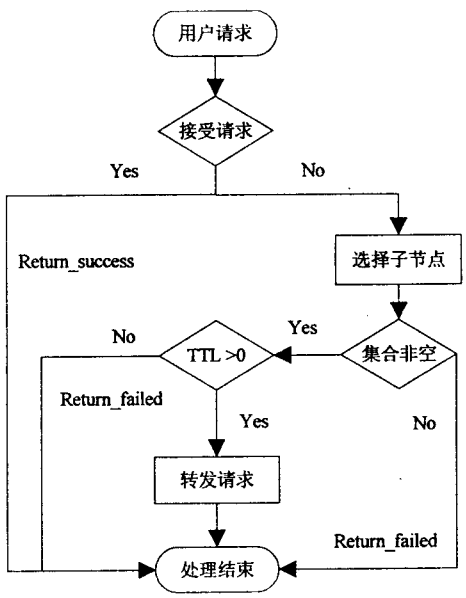


图 6　节点处理流程

4　结束语

针对 P2P 视频点播系统存在的可扩展性以及可靠性等问题,提出了一个视频点播系统的具体设计与实现方案,解决了部分现有 P2PVOD 系统中存在的问题。并对系统在设计与实现中所采用的相关技术与方法作了详细的探讨。随着 IPTV 的提出,基于 P2P 网络模型的流媒体系统将得到更为广泛的应用。

参考文献:

[1] 叶保留,李春洪,姚　键,等.应用层组播研究进展[J].计算机科学,2005,32(6):6-10.

[2] Gao L, Towsley D. Threshold-based multicast for continuous media delivery[J]. IEEE Trans on Multimedia, 2001, 3(4): 405-414.

[3] 刘亚杰,窦文华.P2P 流媒体:一种新型的流媒体服务体系[J].计算机科学,2004,31(4):1-3.

[4] 廖小飞,殷江培,程　斌.基于 P2P 的 VOD 系统中数据缓存策略研究[J].华中科技大学学报,2006,35(8):67-71.

[5] Guo Y, Suh K, Kurose J, et al. P2Cast: P2P patching scheme for VoD service[C]//In: Proc. of the WWW 2003. New York: ACM Press, 2003: 301-309.

[6] Do T, Hua K, Tantaoui M. P2VoD: Providing fault tolerant video-on-demand streaming in peer-to-peer environment [C]//In: Proc. of the IEEE ICC 2004. Paris: IEEE Communications Society, 2004: 1467-1472.

[7] Guo Y, Suh K, Kurose J, et al. A peer-to-peer on-demand streaming service and its performance evaluation [C]//In: Proc of the IEEE ICME 2003. Maryland: IEEE Computer Society, 2003: 649-652.

[8] Hefeeda M, Bhargava B. On-demand Media Stream Over the Internet[C]//Proc of 9th IEEE Workshop on Future Trends of Distributed Computing System (FTDCS'03). San Juan, Puerto Rico: [s.n.], 2003.

[9] 刘亚杰,窦文华.一种 P2P 环境下的 VoD 流媒体服务体系[J].软件学报,2006,17(4):876-884.

[10] 郑常熠,王　新,赵　进,等.P2P 视频点播内容分发策略[J].软件学报,2007, 18(11):2942-2954.

(上接第 16 页)

4　结束语

给出了 Object-Z 类中操作模式拆分为 enable 和 effect 模式的方法和具有继承关系的 Object-Z 类的面向对象程序依赖图,并依此进行切片,最后给出对于具有一个类或具有继承关系的多个类的规格的刻划,通过切片前后 Kripke 结构 K 和 K' 满足 K' 是在公平事件和原子命题之上的 K 的映射来保持验证性质。把具有继承关系的 Object-Z 类整体作为一个模块单独切片,该方法可以扩展到整个 Object-Z 项目,以实现对软件需求规格的验证。

参考文献:

[1] 缪淮扣.软件工程语言-Z[M].上海:上海大学出版社,1999.

[2] Smith G. The Object-Z Specification Language[M]. Hingham, Massachusetts: Kluwer Academic Publisher, 2000.

[3] 吴方君,徐升华.用 Z 形式化描述程序切片[J].小型微型计算机系统,2007,28(8):1444-1447.

[4] 易　彤.前向切片与后向切片之间关系的研究[J].计算机工程与应用,2008,44(12):42-44.

[5] 单卓为,鱼　滨.基于 SPIN 的 CSCW 系统的验证[J].计算机技术与发展,2008,18(4):9-15.

[6] Wu Fangjun, Yi Tong. Slicing Z Specifications[J]. ACM SIGPLAN, 2004, 39(8): 39-48.

[7] Brückner I, Wehrheim H. Slicing Object-Z specifications for verification[J]. LNCS, 2005, 3455: 414-433.

[8] 文志诚,缪淮扣,孙军梅.基于 Object-Z 多态推理[J].计算机科学,2006,33(7):230-233.