

一种基于 P2P 的视频点播系统的研究与设计

张伟文,金鑫,吴国新

(东南大学网络与通信集成教育部重点实验室,江苏南京 210096)

摘要 视频点播(VOD)是一种以用户为主导的音视频信息系统。传统的 VOD 系统多是 C/S 模式采用服务器机群或者代理缓存的方式来实现,这会带来较大的硬件和管理开销,系统扩展性也难以得到保证。文中提出了一种基于 P2P 结构的视频点播的设计方案,分析了采用 P2P 技术以后视频点播系统对于节点闲散资源的利用和保证播放质量等方面的优势,并探讨了在系统设计与实现中的相关技术与方法,包括基于本体的语义搜索、媒体资源定位与节点负载均衡三方面。

关键词 视频点播;P2P;语义 Ontology;负载均衡

中图分类号 :TP39

文献标识码 :A

文章编号 :1673-629X(2007)02-0159-05

Research and Design for VOD System Based on P2P Technology

ZHANG Wei-wen, JIN Xin, WU Guo-xin

(Ministry Edu. Key Lab. of Computer Network and Infor. Integration, Southeast Univ., Nanjing 210096, China)

Abstract Video-on-demand(VOD) is an audio and video integrated information system depends on users' interest. Traditional VOD systems apply client-server mode and are implemented with server-cluster or proxy method. It would bring great hardware and administration cost with low systems scalability. Presents a specific design of VOD system which is based on peer-to-peer technology and analyzes the advantages on using node's resources more effectively and guaranteeing playback quality over those based on client-server architecture. It also includes various technologies used in system implementation such as semantic search, resource location and load balance.

Key words video-on-demand; P2P; semantic ontology; load balance

0 引言

视频点播(Video-On-Demand)是基于流媒体技术而实现的一种网络多媒体系统,它满足了人们自主收看节目、获取视频信息的需求。随着 Internet 的不断发展,一方面,主干网和接入速率的不断提高为各种宽带应用提供了良好的硬件基础设施;另一方面,越来越多的人愿意交互式地观看并分享音视频节目,这些都为视频点播技术的发展创造了条件。目前绝大多数 VOD 系统都是采用 C/S 模式^[1],这样视频服务器的负载非常大,通常靠服务器机群来解决,从而导致较大的成本与管理开销。假如不考虑服务器处理能力的限制,单从网络带宽方面来分析:假设服务器提供 100M 的带宽,每个视频流为 500k,最多也只能同时服务 200 个请求。用户可能有这样的体验,想观看热门的视频非常困难,或者播放时会出现停顿、跳帧,甚至无法继续播放。究其原因,是因为 C/S 模式下每一个用户请

求都与服务器建立单独的通道。这样,即使再强大的服务器也只能承受有限数量的用户,而且用户数量增多后很难保证每个客户端的服务质量。所以,如果要彻底改善视频点播系统的性能,需要提出一种有别于 C/S 的系统结构。

1 相关技术背景

1.1 P2P 技术的兴起

不同于传统的 C/S 模式,P2P 中每个节点的地位都是对等的,也就是说节点既充当服务器为其他节点提供服务,也接受其他节点提供的服务。其网络结构可以分为非结构化和结构化两种。

非结构化 P2P 网络模型主要有:

(1)集中式定位模型:需要一台或多台有特殊用途的服务器为对等点提供索引服务,如 Napster^[2];

(2)分散式定位模型:该模型中的节点只知道直接与它通信或预先定义的那些节点,通过它们参与大型网络中,如 Gnutella^[3];

(3)组播式定位模型:类似于分散式定位,该模型利用网络自身提供的特性来定位和确认节点和资源,

收稿日期 2006-04-24

作者简介 张伟文(1982-),男,江苏苏州人,硕士研究生,研究方向为 P2P 网络技术、网络流媒体;吴国新,教授,博士生导师,研究方向为企业信息化关键支撑技术、计算机集成制造、P2P 网络技术。

使用 IP 组播来实现查询,如 JXTA^[4]。

结构化 P2P 网络模型有很多,其中比较著名的是基于 DHT 的分布式结构化 P2P 模型。这种模型避免了类似 Napster 的索引目录中央服务器,也不像 Gnutella 那样基于广播的泛洪式查找,而是通过分布式 Hash 函数,将输入的关键字唯一映射到某个节点上,然后通过某种路由算法找到该节点并与它建立连接。这种发现机制的典型代表如 Chord^[5]。

1.2 基于本体(Ontology)的语义搜索技术

“本体”一词来源于哲学,在知识共享领域,本体一词用于“概念的规范和表示”^[6],进而应用于资源表示及其之间关系的描述。本体的出现,使得一个领域(领域的范围可以是特定应用中,也可以是更广的范围)内部不同主体(包括资源等)之间进行交流(对话、互操作、共享等)有了一个语义基础,即由本体提供一种共识。这种语义基础通常是用类(Classess)或概念(Concepts)、关系(Relations)、函数(Functions)、公理(Axioms)和实例(Instances)这五种基本建模元素来表征的^[7]。类或概念表示对象的集合;关系表示领域中概念之间的交互作用(如语义关系);函数是一种特殊的关系;公理表示永真断言。由这些概念、关系和实例可以形成清晰的概念层次结构和丰富的语义推理知识,因此,在 P2P 应用中基于本体的资源搜索正成为广泛研究的热点。由于基于本体的方法是基于知识的、语义上的匹配,在查准率和查全率上有更好的保证。

1.3 流媒体播放技术

传统的网络传输音视频等多媒体信息的方式是完全下载后再播放,下载常常要花数分钟甚至数小时。而采用流媒体技术,则可以实现音视频信息的边传输边播放。为了避免因带宽抖动影响播放的质量,基于流媒体的播放往往采用缓存技术予以辅助,也即播放前需要几秒或十几秒的启动延时。流媒体实现的关键技术是流式传输,实现流式传输有两种方法:实时流式传输(Real time streaming)和顺序流式传输(Progressive streaming)。实时流式传输是指保证媒体信号带宽与网络连接匹配以使媒体可被正常播放,顺序流式传输是顺序下载,在下载的同时用户可以观看媒体,它不像实时流式传输在传输期间根据用户网络状况做调整。典型的流媒体解决方案有微软公司的 Windows Media^[8]技术、RealNetworks 公司的 RealMedia^[9]技术和 Apple 公司的 QuickTime^[10]技术。

2 关键技术研究

2.1 视频点播与 P2P 的结合

随着计算机硬件性能的提高和制造成本的下降,

普通 PC 机的处理能力和存储能力已经今非昔比,它们具有一定的闲散资源,诸如计算资源、存储资源和网络资源等。而 P2P 技术正是提供了一种有效使用这些闲散资源的方法。借助 P2P 技术,通过资源分片、节点缓存并服务其他节点的方法可以有效地解决传统视频点播方案中服务器存储和网络带宽的瓶颈。采用节点缓存并服务其他节点的方法具有以下优势:首先,这种方法是基于应用层的,不需考虑下层网络,这种方法中不需要引入专门的服务器或者是“边界代理”,节点只需在回放的同时缓存媒体的一部分数据,由于各个节点开始接收媒体的时间存在差异,所以缓存的数据并不相同,一个节点可以从其他节点中获得所需的数据而保证连续播放,从而这些节点形成一个流传输覆盖。其次,由大量这些节点构成的流传输覆盖的性能不会随着节点数量的增多而下降,并且比 C/S 模式中媒体资源提供者的带宽开销要小,因为节点在消耗其他节点带宽的同时也在贡献自己的部分带宽。再次,每个节点都可以根据一定的 QoS 要求来选择源节点,当覆盖网络中有多个能够提供所需资源的节点的时候,节点可以选择其中性能最好(例如延迟最低)的节点来保证服务质量。

假设媒体资源 R 被分成若干片,最初由节点 C_0 提供。为由节点 $C_i (i = 0, \dots, m)$ 发出的请求定义两个属性: T_i 表示开始接收资源的时间, O_i 表示开始接收时相对资源开始处的偏移。120 节点缓存的长度用 B_i 表示。一段时间以后节点 C_j 也请求 R 资源,如果满足以下不等式 $(T_i - O_i) < (T_j - O_j) < (T_i - O_i) + B_i$, 那么 C_j 就能从 C_i 处获得数据了。节点 C_i 能够向其他节点提供的分片资源数取决于它的上行带宽,对于不能由节点缓存来提供的分片,可以由原始节点 C_0 来提供。图 1 是这种利用节点缓存和带宽资源机制的示意图。图中资源被分成 3 片,假设每个节点缓存 5 分钟的媒体数据并都是从头开始播放,节点旁边的时间是开始播放的时刻,箭头表示可能的数据流向。第一个节点 C_1 直接从源节点 C_0 接收两个分片,第二个节点 C_2 可以从 C_1 处接收 2 个分片,第 3 分片只能从 C_0 处接收,第三个分片可以选择从 C_0, C_2 处同时接收。

这样,从全局来看基于 P2P 的视频点播系统一改 C/S 模式下一个服务节点服务 n 个客户节点的模式,成为多个节点服务一个节点的模式,并且不存在原来意义上的服务器节点,这样既使得全局网络带宽得以有效利用,又能保证单个节点媒体播放的流畅性,而且对于热门资源,观看的人越多,资源越容易获得,播放质量也更能得到保证。

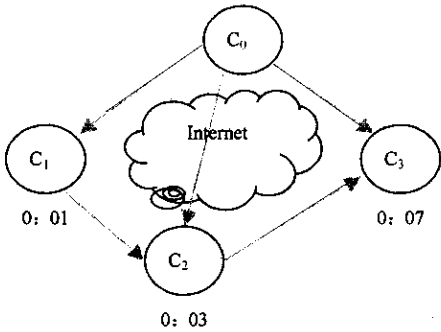


图 1 分片传输示意图

2.2 影片资源的语义描述

目前的视频 VOD 方案中的资源搜索基本都是基于片名关键字的查找,很少有加入语义的要素,更不能根据影片之间的关系来得到相关的电影。如果加入语义功能后会出现这样的使用场景:如果用户查找资源 A,返回的结果不仅仅是 A,还会有与 A 语义相近的其他资源,而这些资源可能也是用户希望查找但是记不起片名关键字的资源。所以如果能设计一种带有语义功能的搜索机制必将改善用户的搜索体验,最大程度地契合用户的实际需求。为此,引入基于本体的语义搜索技术,本体的作用在于使用统一的方式描述资源,并根据资源的相关性搜索出与查询对象密切相关的资源。此处的关键是影片资源的表示和影片与影片之间关系的建立,图 2 示意了一种可行的影片表示和关系本体。

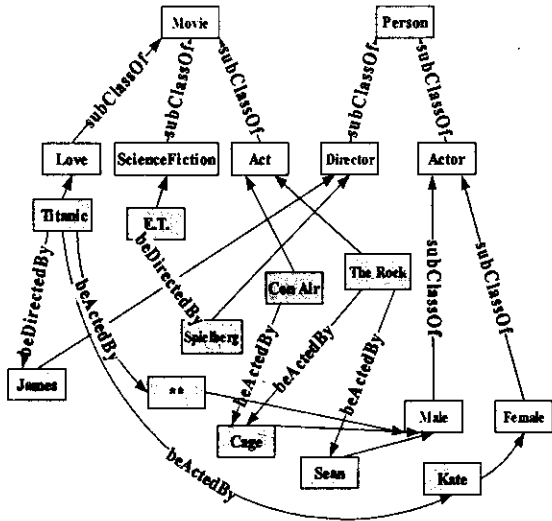


图 2 适于本系统的一种 Ontology 树

图中空心框表示的是类,灰色框表示的是实例,箭头表示定义的关系。通过对影片资源信息的抽象和影片之间关系的定义,为后续的搜索和推理打下了基础。

3 基于 P2P 的视频点播系统的设计

3.1 视频点播系统功能设计

一个视频点播系统要为影片资源的发布、搜索、播

放、管理提供完全解决方案,具体功能设计如下:

- (1)发布功能:节点将自己拥有的影片信息通过某种方式发布给其他各节点。
- (2)搜索功能:根据用户输入的篇名、影片类型等信息返回最匹配的影片以及相关影片信息。
- (3)播放功能:与影片提供者建立连接,通过流媒体传输获取数据并播放。
- (4)管理功能:负责系统的负载均衡,管理影片资源分片,保证系统的可扩展性。

3.2 视频点播系统的结构模块设计

设计方案中的实体如下:

- * 节点。这是所有加入该视频点播系统的节点的集合,任何一个节点都提供一部分的存储和带宽资源。
- * 源节点。源节点是提供某一媒体资源所有分片的节点,也就是某一资源的发布节点,它负责管理系统中当前这一资源所有分片的分布情况。
- * 媒体资源。表示系统中当前共享的媒体资源的集合,每一个资源都有一个唯一的 ID 值用以标识,并用一些特定的关键字来描述。假定每一个资源都用特定的码率编解码,并等分成若干分片,这个分片是节点所能缓存的最小单元。

为了实现前述的各个功能,可以把系统划分为四个功能模块:

(1)底层路由、查找模块的设计。

这一层的功能是维护底层的路由信息的正确性,并且根据上层的资源关键字进行快速准确的定位。对上层提供的接口是接受资源关键字 key 值,返回 key 值所在节点的地址。采用基于 DHT 的分布式路由查找算法 Chord 的改进算法来完成这一功能。Chord 的设计目标就是一个查询原语,追求简单高效、可扩展性好。但是 Chord 也存在着影响查询效率的不足。第一,Chord 没有考虑 IP 层网络的状况(物理位置、网络状况等);第二,系统健壮性不足,节点失效后引起较大的通信开销;第三,稳定机制耗费大量资源。所以考虑对 Chord 算法做如下改进:

- a. 保留多个后继节点,以便增加系统的健壮性。
- b. 路由表每一行增加本节点到后继节点的 RTT 值。
- c. 采用灵活的稳定机制,Chord 算法要定期更新自己的前驱节点和后继节点^[5],但考虑到定期更新会消耗大量的网络带宽,所以考虑只有节点连续 n 次发现不能成功提交任务时,才发起路由表更新。
- d. 为提高查找效率,当节点 N 发布文件 M 时,在路由中遇到的各点上保存 N 的 IP 地址及 M 的标识

符,形如<M,N>,以便查询M时,一旦遇到保存有<M,N>的节点,则能直接得到N的地址。

改进以后的算法在继承 Chord 的优点的基础上更增强了系统的健壮性,使路由更加快速有效,并且在保证正确性的同时减小了系统开销。这为上层的搜索和播放奠定了良好的基础。

(2) 上层语义搜索、查询模块设计。

语义搜索的进行需要一个尽可能丰富的资源知识库,而上述为提高查找效率,节点会保存路由中遇到的资源的相关信息;另外,根据节点多后继节点的设计,以及稳定机制的存在,节点与后继节点之间 Stabilization^[5]的过程中可以互相交换已知的影片信息。随着使用时间的增加,节点上保存的影片信息会越来越丰富,这就保证了语义搜索的效果。影片的信息和关系可以用 XML 表示出来,如:

```
<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns="http://www.owl-ontologies.com/unnamed.owl#"
  xml:base="http://www.owl-ontologies.com/unnamed.owl">
  <owl:Ontology rdf:about="" />
  <owl:Class rdf:ID="Actor">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="Person"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Director">
  <rdfs:subClassOf rdf:resource="#Person"/>
</owl:Class>
<owl:ObjectProperty rdf:ID="beActedBy">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#AnnotationProperty"/>
</owl:ObjectProperty>
<Director rdf:ID="James">
  <Love rdf:ID="Titanic">
    <beActedBy rdf:resource="#D"/>
    <beActedBy rdf:resource="#L"/>
    <beDirectedBy rdf:resource="#James"/>
  </Love>
  . . . . .
```

节点构建一个数据库,存放已知影片的信息,这样便于影片资源的维护,对于一次语义搜索请求,根据提

供的部分影片信息比如片名、演员或者导演等,到 XML 描述中查询出那些其他相关影片也会具有的属性,然后从数据库中根据这些属性,即先前定义的影片之间的关系查询出其它影片的键值(key),进而用这个键值借助底层路由结构定位影片资源的提供者。

(3) 负载均衡模块的设计。

随着加入网络的节点的增多,对资源的访问往往会簇集化,即某些节点上的资源请求会超过自身的服务能力而变成热门节点,而路由到这个节点的路径(类 Chord 环上该节点逆时针方向上的若干节点构成的路径)会变成热门路径。对于热门资源和热门路径的存在,如果不采取措施,会导致该资源路由的暂时性失效。针对上述问题,文中提出几点节点负载均衡的方法。

① Chord 算法在指定节点 ID 和资源 key 值时采用一致性 Hash 函数,这已经有了 key 值在 Chord 环上均衡分布的考虑,但是这还不够。考虑在指定节点 ID 的时候,不仅仅 Hash 可以唯一表示节点的值而得到一个 key 值,还考虑引入“slot”的概念^[11]。slot 是指节点所能取的一系列 key 值: $h(ip_1), h(ip_2), \dots, h(ip, c \log n)$, h 表示 Hash 函数, c 是常量。对于 slot 中的每一个 key 值,考虑紧邻并小于该 key 值的节点 key 值为 N_i^- , 紧邻并大于该 key 值的节点 key 值为 N_i^+ , 那么节点在加入系统的时候最终选择的 key 值 $h(i, j)$ 满足在下列 N_j^- 与 N_j^+ 之间: $N_j^+ - N_j^- = \max\{N_j^+ - N_j^- \mid 1 \leq j \leq c \log n\}$ 。

② 多套 Hash 函数的引入。采用多哈希函数或关键字加索引的方式把每个文件索引信息映射到 k 个节点,增强文件的可用性和查找效率,也使查询负载得到一定均衡,缓解“热路径”现象。

③ 对于解决“热路径”问题,除了前面提到的发布文件时节点保存<资源,节点>对,还可以考虑在路由表中增加“虚拟节点”的若干前继的方法并构建适当策略来缓解“热路径”问题^[12]。如果一个节点发现由于其他节点请求其上的某个资源而导致负载超过了一定限额,那么节点可以把这个资源的部分或全部 push 到与之相邻的节点上,可以是前继或后继节点,或者其他有较多剩余资源的节点(根据路由表中的 RTT 值和当前负载情况),并且由它来负责这份被 push 的复本的有效性。由于是影片资源,所以一致性的问题可以不考虑(文件内容不会变动)。

(4) 影片传送和播放模块的设计。

这一部分主要负责影片分片的管理、传输和播放。影片资源分片流式化,并通过网络连接传输到请求节点的缓存中。由于是多个节点同时为一个节点服务,

所以缓存中的分片的到达并不一定是按照播放顺序的,所以要它们进行重新排序和拼接。然后通过开发的过滤器^[13]进行播放,显示在客户端界面上。这个模块的实现可以采用 Microsoft 的 DirectShow 技术^[13]。各模块之间的关系如图 3 所示。

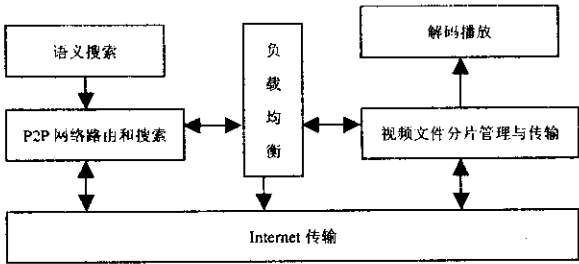


图 3 系统结构图

4 视频点播流程分析

跟踪一次具体的播放流程,如图 4 所示。

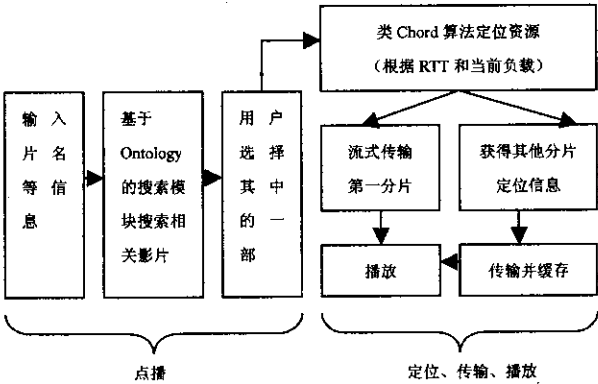


图 4 本 VOD 系统点播到播放过程图

5 结束语

视频点播系统(VOD)是当前高带宽网络上一个很有前景的音视频信息服务系统。针对传统 C/S 模式下 VOD 系统的不足,提出了一种基于 P2P 网络的视频点播系统的具体设计与实现方案,对实现过程中的相关技术和方法作了一定探讨。它具有搜索效果

好、查找定位资源迅速、节点资源可用性强、系统可扩展性好的特点。相信随着计算机网络带宽的进一步提高和计算机性能的增强,视频点播会越来越广泛地应用在教育、娱乐、办公等领域。

参考文献：

[1] 沈建强,耿兆丰,邹 轩. 视频点播系统的设计与实现[J]. 计算机工程, 2006, 32(2): 209-211.

[2] Stern R. Napster: a walking copyright infringement? [J]. Micro IEEE, 2000, 20(6): 4-5.

[3] Jovanovic M A. Modeling large-scale peer-to-peer networks and a case study of Gnutella [D]. Ohio: University of Cincinnati, 2001.

[4] Li Gong. Project JXTA: A technology overview [R]. [s.l.]: SUN Microsystems, 2001.

[5] Stoica I, Morris R, Karger D et al. Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications [R]. TR-819. [s.l.]: MIT, 2001.

[6] Gruber T R. A translation approach to portable ontologies [J]. Knowledge Acquisition, 1993, 5(2): 199-220.

[7] Gruber T R. Toward principles for the design of ontologies used for knowledge sharing [EB/OL]. Padua workshop on Formal Ontology, 1993-03. <http://ksl-web.stanford.edu/KSL-Abstracts/KSL-93-04.html>.

[8] 陆其明. DirectShow 实务精选 [M]. 北京: 科学出版社, 2004.

[9] RealNetworks, Inc. Realmedia [EB/OL]. 2005. <https://helixcommunity.org/>.

[10] Apple, Inc. Quicktime [EB/OL]. 2006. <http://developer.apple.com/quicktime/>.

[11] Karger D R, Ruhl M. New Algorithms for Load Balancing in Peer-to-Peer Systems [R]. LCS-TR-911. [s.l.]: MIT, 2003.

[12] 徐家臻. 基于 DHT 方法的 P2P 网络负载均衡研究 [D]. 武汉: 武汉大学, 2004.

[13] Microsoft, Inc. DirectX Media SDK [EB/OL]. 2003. <http://www.microsoft.com>.

(上接第 158 页)

系数据库结合,充分发挥两种数据库的优点,利用工作流引擎 Workflow3.0,实现强大、灵活的工作流配置功能;采用多种技术保证系统的安全性和保密性。

参考文献：

[1] 郑 翔. LotusScript 与 Formula 应用手册 [M]. 北京: 机械工业出版社, 2001.

[2] Loney K. ORACLE 8 数据库管理员手册 [M]. 何江华译. 北京: 机械工业出版社, 2000.

[3] 诺克斯. Oracle Database 10g 安全性高效设计 [M]. 刘桂清,

等译. 北京: 清华大学出版社, 2006.

[4] Riggsby M, McCoy C. Lotus Domino&Notes R5 入门到精通 [M]. 霍明岳译. 北京: 电子工业出版社, 2000.

[5] 袁国刚. 基于 Lotus Notes 的政务办公自动化系统的研究与实践 [J]. 微型电脑应用, 2003, 12(12): 35-37.

[6] Kaschek R, Thalheim B. LotusScript and Statement Construction Rule [J]. ACM SIGMOD Record, 2002, 31(5): 16-18.

[7] Melton J. LotusScript Language Reference [J]. ACM SIGMOD Record, 2003, 20(3): 28-30.