3Tnet 视频点播中媒体分发系统的设计与实现①

Design and Implementation of Media Delivery System In 3Tnet VOD

郑　烃　骆　维　（中国科学技术大学 网络传播系统与控制联合实验室 网络传播系统与控制安徽省重点实验室 安徽 合肥 230027）

摘　要： 根据 3Tnet 中视频点播的特点，在中心内容服务器和传统边缘缓存代理之间设计了一种媒体分发系统。该系统使用 P2P 的方式连接各边缘缓存节点，充分利用 3Tnet 带宽，能有效提高视频内容的分发速度；另外还使用动态链接库技术提供了更为灵活的视频分发调度机制。实验表明，该系统对硬件资源消耗较小，可以直接建立在已有的边缘代理服务器上，无须大量额外硬件投资，适合大规模 IPTV 商业运营。

关键词： 3Tnet　视频点播　P2P　内容分发

1　引言

当前计算机网络和通信技术的高速发展，使得视频点播服务（VOD）成为高速网络中日益重要的服务方式之一。传统 Internet 采用 best – effort 方式进行流媒体服务存在不少问题[1]，例如带宽无法满足流媒体增长需求，QoS 难以保证等。3Tnet[2]，是国家‘八六三计划’的重大专项，它集成了 Tbit/s 级高速光传输、Tbit 级智能光网络（ION）、Tbit/s 级网际协议（IP）路由器和 Tbit/s 级应用支撑环境，能够为视频点播业务提供可靠的 QoS 保证。因此，在 3Tnet 中开展视频点播业务具有很大潜力。

传统视频点播系统一般在中心服务器和客户之间设立代理缓存服务器以提高服务质量，目前主要有两个研究方向：一是改进代理缓存的分发调度策略，以提高本地缓存命中率，减轻中心服务器负担，如文献[3]提出将整个流媒体对象进行缓存，文献[4]提出了指数增长的分段缓存策略。但这些策略都是针对某种特定情况或假设，并不能完全适应现实中复杂的用户点播行为和网络情况。第二个研究方向是改进系统架构，如文献[5,6]都提出了在视频点播系统中引入 P2P 技术，将每个缓存代理以及其下属的用户组成一个自治系统，在自治系统中使用 P2P 技术来提高服务质量。该方法没有充分利用到 3Tnet 中缓存代理之间丰富的带宽，并且 P2P 自治系统实现较为复杂。文献[7]提出了多级缓存的设计，在中心服务器和代理缓存服务器之间设立内容分发平台来减轻中心服务器负担，这样就需要增加大量额外的硬件投资。本文的主要贡献在于，设计了一种能充分发挥 3Tnet 带宽优势，提高视频内容分发速度的媒体分发系统（Media Delivery System）MDS，并且可建立在现有服务器之上，无需大量额外硬件投资。同时，该系统还支持分发调度策略的灵活替换，可以根据具体网络情况和用户点播行为选择最合适的策略进行执行。

2　相关技术

2.1　基于代理缓存的传统视频点播系统

基于代理缓存的传统视频点播系统架构如图 1 所示。

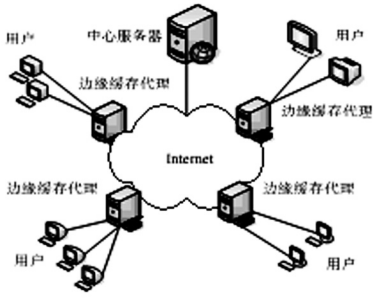

图 1　传统代理缓存结构示意图

① 基金项目：国家发改委项目（CNGI – 04 – 15 – 2A）；国家 863 项目（2005AA103310）；国家 863 项目（2006AA01z114）

主要是在中心内容服务器和客户之间设立代理缓存服务器,把视频内容分发到网络的边缘,用户点播时能从最近的边缘节点得到快速响应,这样就减轻了中心服务器的压力,也提高了系统并发服务数量。由于缓存节点的空间有限,所以一般只完整存储点播率非常高的热门影片,对普通视频数据则采用部分存储的方式。在用户点播时,如果本地的代理缓存服务器存有用户请求的数据,则直接由本地代理向用户提供服务,该情况称为数据命中。如果没有命中,则代理服务器需要向中心服务器请求数据。由于视频数据的访问规律不同于一般文件,热门视频的访问率远大于冷门视频,因此为提高本地数据的命中率,需要根据各种分发调度策略及时调整各代理的缓存数据,使其符合用户的点播行为。

2.2 动态链接库技术

动态链接库是一种通用的软件组件技术。可以事先对程序的函数进行编译,然后将它们存入动态链接库文件中,在程序运行需要使用该函数时才载入,具有很大的灵活性。同时,在不需要动态链接库时,还可以释放其所占用的内存,腾出空间供其它程序使用。因此使用动态链接库的程序代码规模比较小,运行所占用的内存也较少。本文的媒体分发系统选择动态链接库来实现分发调度策略,不仅降低了系统的内存占用,还可以根据实际情况灵活选择最适合的策略。同时,将来要增加新策略时,只需要将该策略编译为库文件,即可动态链接到系统中,立即投入使用,整个系统无须重新修改和编译,也避免了二次开发的成本。

3 媒体分发系统设计和实现

3.1 系统整体框架

系统整体架构如图 2 所示,各模块具体介绍如下:

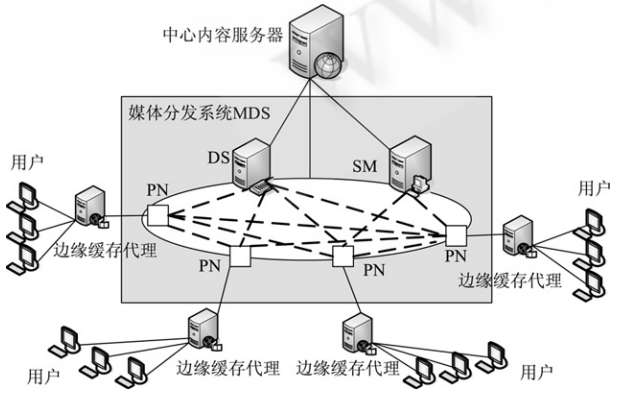

图 2　媒体分发系统整体架构

(1) PN (P2P Node) 媒体分发系统中的 P2P 节点,是边缘缓存代理的 P2P 数据接口,缓存代理通过 PN 进行 P2P 数据发送和接收,PN 负责完成在整个系统中的 P2P 资源查询和数据传输工作。缓存代理和 PN 是一对一的关系,每个边缘缓存代理都对应一个自己的 PN。PN 同时负责定期从缓存代理处获取所有视频的用户点播信息,上报给分发调度策略管理模块。

(2) DS (Directory Server):P2P 目录服务器,记录了整个 P2P 系统的资源分布情况,向 PN 提供 P2P 资源查询服务。一个 PN 在进行数据下载之前,先找 DS 查询所要的数据在哪些缓存代理上有,然后向这些缓存代理对应的 PN 发起数据请求。

(3) SM (Strategy Manager):分发调度策略管理模块,管理并执行整个系统的分发调度策略。该模块定期通过 PN 收集影片在每个缓存代理上的点播情况,根据具体网络情况选择一个合适的分发调度策略,然后计算出视频数据分发调整指令,发送给各 PN,调整各缓存代理的数据存储,使其更符合用户的点播行为,提高本地数据的命中率。

3.2 P2P 数据分发流程

系统的 P2P 媒体数据分发流程如图 3 所示。具体为:

(1) SM 向边缘缓存代理对应的 PN 发出视数据分发指令。

(2) PN 向 DS 查询所需数据在其它哪些缓存代理上。

(3) PN 根据 DS 的返回结果,向有自己所需数据的缓存代理对应的 PN 请求数据,如果所有缓存代理都没有所需数据,则向中心内容服务器请求数据。

(4) PN 将获取到的数据发送给自己的缓存代理。

(5) 数据传输完成后,PN 向 SM 上报分发指令执行结果,同时向 DS 上报自己所对应的缓存代理的数据变化。

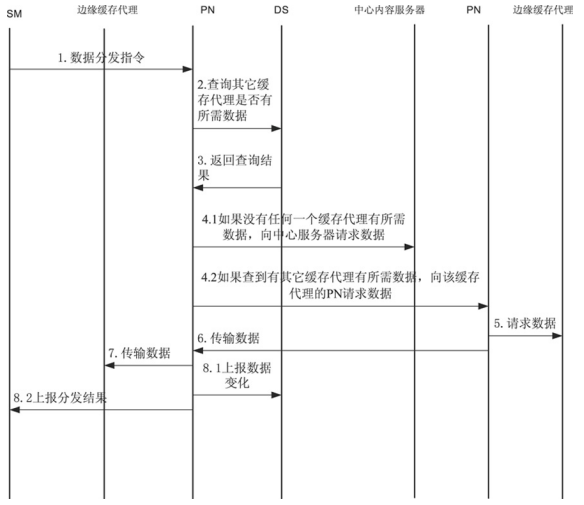

图 3　P2P 数据分发流程

3.3 系统各模块实现

3.3.1 DS

DS 负责维护整个系统的 P2P 资源信息，通过维护一个可用 PN 信息列表来实现，该列表记录了从哪些 PN 处可以获取到哪部影片的数据。列表中最初只有中心内容服务器，随着影片数据被分发到各个缓存代理中，DS 会逐渐添加这些缓存代理对应的 PN 的信息。可用 PN 信息列表格式如表 1 所示。

表 1　可用 PN 信息列表

影片列表	可用 PN 列表		
影片 1	中心内容服务器	PN02	PN05
影片 2	中心内容服务器	PN01	PN03
影片 3	中心内容服务器	PN06	
影片 4	中心内容服务器		

在该表中，记录了当前 4 部不同影片 1，2，3，4 所对应的可用 PN 信息。当 PN 向 DS 查询某部影片的 P2P 资源信息时，DS 返回该影片对应的 PN 信息，包括 PN 的 IP，资源存储情况等。

可用 PN 信息列表需要同步更新，当某部影片被分发到一个新的缓存代理时，PN 会上报该代理的数据变化，DS 立刻更新可用 PN 信息列表，以实现信息同步。如果某个内容从一个缓存代理中删除，该代理对应的 PN 也会将信息变化告知 DS，以更新列表。

DS 同时负责对 PN 进行监测，定期进行测试 PN 的网络状况和处理能力等，如 PN 出现异常会暂时从可用列表内删除，在下一次检测正常后再将该 PN 重新加入可用列表。

3.3.2 PN

PN 在实现时兼顾了性能和系统资源。对数据量占用非常小的 P2P 数据源列表，将其缓存在了内存中，以加快 P2P 资源查询速度。而对于数据量大的视频数据，采用了阻塞式的接收和传输方式，即在把当前数据发送给缓存代理或存入硬盘后，才开始接收下一块 P2P 数据，这样虽然在带宽利用率上会有所下降，但不会在内存中堆积视频数据，大幅度减少了内存占用。

（1）PN 内部实现及主要工作流程

媒体分发系统中各部分内部实现及其工作流程如图 4 所示。PN 维护有自己的数据源缓存列表，存有最近一段时间从 DS 查询到的数据源，当 PN 收到数据分发指令后，就向这些数据源请求数据，接收到数据后转发给缓存代理，如果当前数据源不足以完成任务，则再行向 DS 查询最新的数据源。在任务完成后，向 DS 上报更新后的 PN 情况，向 SM 上报指令执行结果。

（2）P2P 数据源选择

PN 在从 P2P 网络中下载数据时，对数据源的选择非常重要，为减轻中心内容服务器的负担，在实现时优先考虑了从其它 PN 处请求视频数据。对于这些 PN，也参考了该 PN 与自己的网络距离，可用带宽，当前负载情况等参数，然后对这些 PN 进行排序，选择条件最好的 PN 优先下载。在对数据源的选择上，为每个可用的 PN 设置了一个权值 W，将 PN 间网络距离 D 设置为权值 d，可用带宽 B 设置为权值 b，当前负载 L 设置为权值 l。PN 的权值 $W = B \times b - D \times d - L \times l$。中心内容服务器的权值设为最低。请求数据时从数据源中选择权值 W 最大的几个 PN，并发传输数据。具体的权值根据系统实际运行情况行调整，以获得最佳取值。

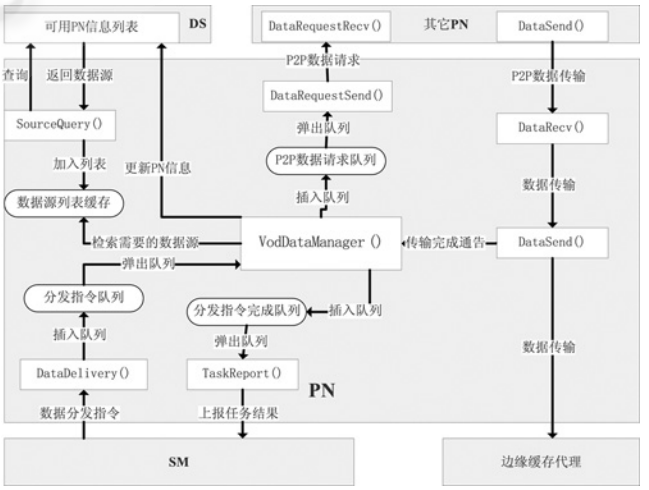

图 4　PN 内部实现

3.3.3 SM

SM 的实现架构如图 5。首先将各种分发调度策略编译为对应的动态链接库文件，然后策略管理模块收集整个系统的视频点播情况和网络情况，选择一个最合适的分发策略，再由 SM 主程序将该策略对应的动态库加载入内存中，调用库中已编译的分发调度函数，计算出对所有 PN 的数据分发调整指令，最后发送给所有 PN，完成分发调度。

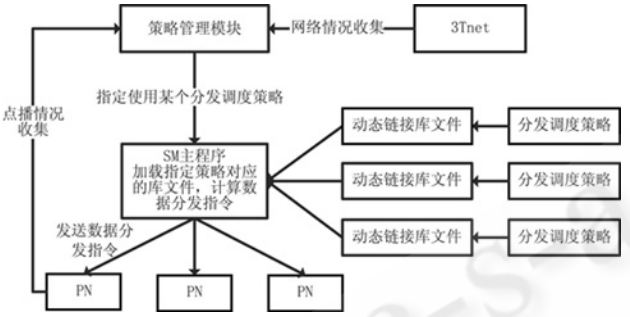

图 5　SM 模块架构

其中 SM 加载动态库的主程序逻辑实现过程如下：

（1）输入分发策略对应的动态链接库文件名

（2）获取当前所有 PN 信息

（3）根据输入的动态链接库文件名，加载指定的调度策略到内存中

（4）输入所有 PN 信息，使用上一步加载的调度策略计算出对每个 PN 的数据分发指令

（5）发送分发指令给所有 PN

如果要新增，或是改变已有的分发调度策略，只需将分发策略编译为动态链接库文件，增加或替换原有库文件即可，SM 程序本身无须做任何修改。

4　系统性能测试

为了验证 MDS 系统的性能，使用了 NS2（Network Simulator，version 2）软件对系统进行媒体分发速度测试。NS2 是目前广泛使用的一种网络模拟软件，由 UC Berkeley 开发而成，本身具有一个虚拟时钟，能产生各种网络仿真事件。测试平台的服务器为 Dell PowerEdge 2650，Intel Xeon 2.4 GHz CPU 双核，4 GB 内存，操作系统为 RedHat_AS4。分别测试了在传统架构和增加了 MDS 情况下，全网缓存代理数从 1 到 500 时，中心服务器把一部影片分发到所有代理上所需要的时间变化。其中，中心影片服务器的出口带宽设置为 1Gbps，各缓存代理的出口带宽设置为 100Mbps，分发影片大小设置为 500MB，时间记录为从第一个缓存代理开始接收数据开始，到最后一个代理接收数据完毕，实验结果如图 6 所示。

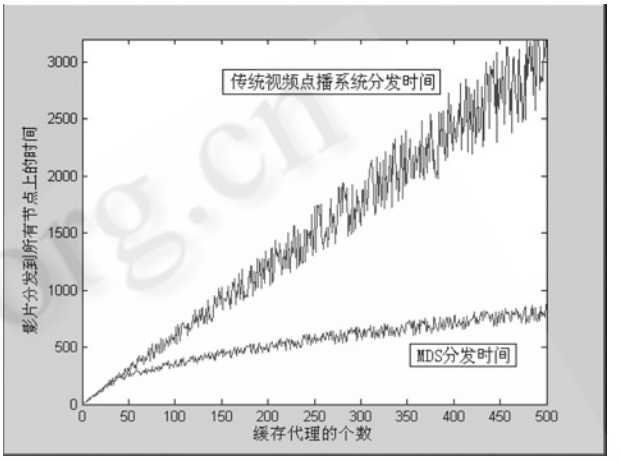

图 6　影片分发时间对比

从实验结果可以看出，在缓存代理数小于 50 的时候，传统架构和 MDS 的分发时间相差不大，但随着代理数的增加，传统架构的分发时间呈线形增长，且波动幅度很大，这是由于中心服务器负担过重，网络情况不稳定导致。而 MDS 的分发时间增长缓慢，且波动幅度很小，网络情况稳定，分发速度明显高于传统架构。

同时对 PN 的单机硬件消耗情况做了测试，结果如表 2：

表 2　PN 单机硬件消耗测试

P2P 数据传输速率	CPU 使用率	内存占用量
10MBps	1.8%	18MB
100MBps	5.2%	34MB
600MBps	8.9%	114MB

从测试结果来看，即使在 600MBps 的 P2P 数据传输量情况下，PN 对 CPU 和内存的消耗也比较低，完全可以运行在现有的代理缓存服务器上而不影响其正常工作。

对 DS 的单机服务能力测试结果为：每秒能提供约 13000 次 P2P 资源信息查询，如果是 500 个代理服务器，则每个服务器每秒能得到 2.5 次服务。而在 NS2 所模拟的 500 个服务器同时分发情况下，每个 PN 平均

每秒向 DS 查询了 0.12 次,远远小于 DS 的服务能力,因此,用一台普通双核服务器运行 DS 即可。

SM 对硬件的要求与具体分发策略有关,但因为使用了动态链接库,不进行使用的策略可以从内存中撤出,所以内存占用不会超过 50M,并且代理服务器数据调整的频率远小于 P2P 资源查询的频率,所以 SM 对硬件资源的消耗大大低于 DS,一台普通服务器即可运行 SM,在服务器数量不大于 100 时甚至可以将 SM 和 DS 运行于同一台服务器上。

综上,整个 MDS 可以运行在传统视频点播系统的分发架构之上,只需增加一到两台普通服务器运行 DS 和 SM,无须其它额外的硬件投资。

5 结束语

本文根据 3Tnet 的特点,设计和实现了一种媒体分发系统(Media Delivery System) MDS,可建立在传统视频点播系统的分发架构之上,无需大量硬件投资,能有效提高视频内容分发速度,并支持分发调度策略的灵活替换。目前系统已经完成了开发和测试,主控设备已部署在上海宽带技术及应用工程研究中心,在 3TNet 上试运营,速度比传统的 C/S 视频分发模式有着明显的提高。

今后的工作是充分利用 MDS 灵活的分发策略替换机制,研究更适合 3Tnet 视频点播的分发调度策略,进一步提高 MDS 的工作效率。

参考文献

1 Hofmann M, Ng E, Guo K, et a1. Caching techniques for streaming multimedia over the Internet[R]. Bell Laboratories Technique Report, BL01 1345 - 990409 - O4TM ,1999.

2 高性能宽带信息示范网(3Tnet)专项课题指南(2004). http://www.863.org.cn/863_105/applyguide/guide_infotech/200406010039.html

3 Almeida J M, Eager D L, Ferris M, et a1. Provisioning content distribution networks for streaming media. Proceedings of IEEE INFOCOM. 2002,3:1746 - 1755.

4 Wu K L, Yu P S, Wolf J L Segment - based proxy caching of multimedia streams. Proceedings of International WWW Conference. Hongkong, 2001:36 - 44

5 杨传栋,余镇危,王行刚. 结合 CDN 与 P2P 技术的混合流媒体系统研究. 计算机应用, 2005,25(9):2204 - 2207.

6 邓亮. P2P 与 CDN 结合实现 IPTV 点播业务. 网络应用,2007,(1):39 - 44.

7 向伟,李俊,吴刚,陈卿. 基于 3Tnet 的视频点播服务策略. 中国科学技术大学学报, 2007, 37(2):189 - 194.