

基于 P2P 的 VOD 系统中的数据传输分配算法

杨传栋¹, 余镇危², 王行刚³, 张燕妮²

YANG Chuan-dong¹, YU Zhen-wei², WANG Xing-gang³, ZHANG Yan-ni²

1. 山东农业大学 信息技术与工程学院, 山东 泰安 2710182

2. 中国矿业大学 机电与信息工程学院, 北京 1000832

3. 中国科学院 计算技术研究所, 北京 100080

1. Department of Information Engineering, Shandong Agriculture University, Taian, Shandong 271018, China

2. Department of Electronic and Information Engineering, China University of Mining & Technology, Beijing 100083, China

3. Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100080, China

E-mail: ycdong@sdau.edu.cn

YANG Chuan-dong, YU Zhen-wei, WANG Xing-gang, et al. Data allocation algorithms in P2P based VOD. Computer Engineering and Applications, 2007, 43(35): 150-152.

Abstract: In P2P based VOD system design, how to assign media data to multiple supplying peers in one streaming session is a core problem. This paper presents an optimal media data assignment algorithm ODTA_{P2P}, which results in minimum buffering delay in the consequent streaming session. The proof of the optimality of the algorithm is given, and the efficiency is tested by experiments. The time complexity of the algorithm is similar with other known task assignment algorithm, while it can suitable in more common environment where no any special assumption.

Key words: P2P; media streaming; VOD; task assignment; buffering delay

摘要: 在基于 P2P 的 VOD 系统中, 流媒体数据的传输模式一般是多对一的, 数据传输任务在多个源节点间的分配是系统设计的一个核心问题。在保证流媒体连续播放条件下, 给出了一个以缓冲延迟为优化目标的最优化分配算法 ODTA_{P2P}, 并对算法的最优性给出了数学证明, 并通过实例验证了算法的有效性。该算法在计算复杂性与同类算法相同的前提下更适合于实际环境下应用。

关键词: P2P; 流媒体; VOD; 任务分配; 缓冲延迟

文章编号: 1002-8331(2007)35-0150-03 **文献标识码:** A **中图分类号:** TP393

1 引言

在基于 P2P 的 VOD 系统中, 对等节点(Peer)既可以点播其他对等节点上存放的流媒体内容, 也可以将自身缓存的流媒体内容提供给其他节点点播, 即每个对等节点既是客户端也充当流媒体服务器的角色^[1,2]。系统中的对等节点是一些普通的计算机节点, 具有以下几个特性^[3]:

(1) 与传统的流媒体服务器节点相比, 其所能够或愿意提供的带宽资源有限, 而流媒体数据率较高, 因此通常需要多个节点才能为单个节点提供数据服务;

(2) 不同对等节点的带宽资源各不相同, 具有异构性;

(3) 各对等节点加入或退出系统的时间是随机的而且难以预测;

(4) 各对等节点在下载播放流媒体内容的同时也将这些内容在本地缓存以供其他节点点播, 因此系统的服务能力是逐渐增长的, 假如系统的节点越多, 其服务能力越强。

在设计基于 P2P 的 VOD 系统时需要解决以下一些问题:

(1) 当多个对等节点都缓存有用用户的请求内容时, 应如何

选择服务节点;

(2) 正在提供服务的对等节点突然退出系统时, 如何保证服务接受者的 QOS 尽量不受影响;

(3) 对等节点(有缓存能力的客户节点)上的缓存替换问题;

(4) 流媒体数据传输的任务分配问题, 即当多个服务提供者为一用户节点提供服务时, 各提供者之间的传输任务应如何分配。

对 P2P 流媒体数据传输的任务分配问题, 文献[4]研究了分层编码流媒体的情况, 针对 CBR 流媒体的情况, Xu Dong-yan 等人在文献[3]中提出了一种分配算法 OTS_{P2P}, 该算法在假定各服务提供节点的输出带宽可表示为 $R_0/2^k$ 且总和恰好为 R_0 时(R_0 为流媒体正常播放时的数据率)可获得最优分配结果, 即在保证流媒体连续播放的前提下, 可使缓冲延迟达到最小。本文的工作是针对 CBR 流媒体数据传输的任务分配问题, 给出一种最优数据传输分配算法 ODTA_{P2P}, 并对算法分配结果的最优性从理论上给出证明。

2 目标问题与算法描述

2.1 问题描述

在基于 P2P 的 VOD 系统中,流媒体文件被划分为小的等长的数据块,其传输调度以块为基本单位。并假定媒体流是 CBR(Constant Bite Rate)的,因此每一块的播放时间都是等长的。

当一个节点请求播放某个流媒体文件时,首先调用 P2P 网络的资源服务,发现可能的资源供应节点,并通过一定的策略选择实际的资源供应节点,然后调用传输任务分配算法决定各供应节点应传输的数据段,各供应节点收到传输任务后便依照播放的先后顺序将其应发送的数据块依次发送给请求节点,请求节点一边接收一边播放传输过来的流媒体数据,并将这些数据缓存在本地磁盘上,一旦播放结束,该请求节点便可以将这些缓存的数据提供给其他节点,从而成为一个服务提供者^[1]。

记请求节点 P_r 的下行带宽为 B_0 , 其供应节点 $P_1, P_2, P_3, \dots, P_n$ 的上行带宽为 B_1, B_2, \dots, B_n , 一般情况下,各供应节点同时启动数据传输任务,因此应有:

$$B_0 \geq \sum_{i=1}^n B_i$$

为保证流媒体播放的连续性,请求节点在收到流媒体数据后并不立即播放,而是先将数据存入缓存中,当缓存中的数据达到一定量之后才开始播放。从开始数据传输到开始播放之间的这段时间间隔称为缓冲延迟。为了减少客户的等待时间,缓冲延迟应尽可能的短。

给定一种传输任务分配方案,不考虑网络变化的情况下,引理 1 给出了在保证流媒体连续播放的前提下,其最小缓冲延迟的确定方法。

引理 1 对于一个给定的数据传输分配方案,假设数据传输启动的时刻记为 0,按播放顺序,记各数据块传输完成的时刻为 t_1, t_2, \dots, t_n , 设每个数据块的播放时长为 Δt , 则该分配方案保证流媒体连续播放的最小缓冲延迟 D 由下式给出:

$$D = \max_{i \in \{1, 2, \dots, n\}} \{t_i - (i-1)\Delta t\}$$

证明 为保证流媒体播放的连续性,每个媒体块在被播放时必须已传输完成,因此下列各式必须同时成立:

$$D \geq t_1, D + \Delta t \geq t_2, D + 2\Delta t \geq t_3, \dots, D + (n-1)\Delta t \geq t_n$$

即 D 为同时满足下列条件的最小数值:

$$D \geq t_1, D \geq t_2 - \Delta t, D \geq t_3 - 2\Delta t, \dots, D \geq t_n - (n-1)\Delta t$$

因此 D 取值为:

$$\max_{i \in \{1, 2, \dots, n\}} \{t_i - (i-1)\Delta t\}$$

即:

$$D = \max_{i \in \{1, 2, \dots, n\}} \{t_i - (i-1)\Delta t\}$$

证毕。

数据传输任务分配算法的作用就是给出一种数据传输的分配方案,该方案的缓冲延迟在所有的分配方案中最小。即对一个请求节点和一组供应节点,分配算法决定各供应节点应传输那些文件块,优化目标是在保证流媒体连续播放的前提下缓冲延迟最小。

2.2 算法描述

算法的基本思想是按播放顺序依次分配各数据块,在前 $i-1$ 块已分配完成的基础上,将 i 分配给能最早下载完成 i 的节点。为了叙述方便,我们将该算法记为 ODTA_{P2P}, 算法的详细描述如下,其中 B_i 表示节点 P_j 的上行带宽, L 为数据块的长度,最小缓冲延迟有变量 D 给出。

```

ODTA_P2P( $P_1, P_2, \dots, P_m, D$ ){
    For( $j=1; j \leq m; j++$ ){
         $T[j]=0$ ;
    }
     $D=0$ ;
    For( $i=0; i \leq n; i++$ ){
        For( $j=1; j \leq m; j++$ ){
             $Temp[j]=T[j]+L/B_j$ ;
        }
         $s=1$ ;
         $Tmin=Temp[1]$ ;
        For( $k=2; k \leq m; k++$ ){
            If( $Temp[k]<Tmin$ ){
                 $Tmin=Temp[k]$ ;
                 $s=k$ ;
            }
        }
         $i \rightarrow P_s$ ; //将  $i$  分配给  $P_s$ 
         $T[s]=Tmin$ ;
        If( $D < T[s] - i \cdot \Delta t$ ){
             $D = T[s] - i \cdot \Delta t$ ;
        }
    }
}
    
```

2.3 算法的最优性证明

定义 1 如果一个分配方案 A 的最小缓冲延迟不大于另外一个分配方案 B , 称分配方案 A 不差于分配方案 B 。

定义 2 如果分配方案 A 的最小缓冲延迟不大于任意一个分配方案, 称分配方案 A 为一个最优分配方案。

定义 3 对一个分配方案, 如果各数据块传送完成的时间顺序与播放顺序一致, 则称该分配方案为正规分配方案。

下面证明 ODTA_{P2P} 算法得到的分配方案为最优分配方案, 为此先证明两个定理。

定理 1 对一个分配方案, 假设数据传输启动的时刻记为 0, 按播放顺序, 记各数据块传输完成的时刻为 $t_0, t_1, t_2, \dots, t_n$, 若存在两个连续的数据块 $i, j (j=i+1)$ 满足 $t_i \geq t_j$, 则交换 i, j 的分配后得到的新的分配方案不差于原来的分配方案。

证明 因为同一服务节点分配的数据块的传输完成时间与播放顺序是一致的且不会相等, 因此, 由 $t_i \geq t_j$ 可知 i, j 必定分配给了不同的两台服务节点 P_i 与 P_j 。

交换 i, j 的分配, 即将 j 分配给 P_i 将 i 分配给 P_j , 形成一个新的分配。

由于 i 和 j 在播放时为连续块, 因此在分配给 P_i 的各块中原来比 i 早传输的仍然要比 j 早传输, 原来比 i 晚传输的也仍然比 j 要晚传输, 即 j 替代 i 分配给 P_i 后, j 在 P_i 的传输次序为原来 i 的传送次序, 且对其它各块无任何影响。

同理 i 替代 j 分配给 P_j 后, i 在 P_j 的传输次序为原来 j 的传送次序, 且对其它各块无任何影响。因此该新分配的各数据块传输完成的时刻为:

$$t_0, t_1, \dots, t_j, t_i, \dots, t_n$$

即 i 块的传输完成时刻为 t_j, j 块的传输完成时刻则为 t_i , 其余不变。

由引理 1, 原来分配方案的最小缓冲延迟为:

$$D = \max\{t_0, t_1 - \Delta t, \dots, t_i - i\Delta t, t_j - j\Delta t, \dots, t_n - n\Delta t\}$$

而交换 i, j 后的新分配方案的缓冲延迟为:

$$D' = \max\{t_0, t_1 - \Delta t, \dots, t_j - i\Delta t, t_r - j\Delta t, \dots, t_n - n\Delta t\}$$

由 $i < j$ 且 $t_i \geq t_j$ 可知

$$t_j - i\Delta t \leq t_r - i\Delta t$$

且

$$t_r - j\Delta t \leq t_r - i\Delta t$$

从而:

$$\max\{t_0, t_1 - \Delta t, \dots, t_j - i\Delta t, t_r - j\Delta t, \dots, t_n - n\Delta t\} \leq \max\{t_0, t_1 - \Delta t, \dots, t_r - i\Delta t, t_r - j\Delta t, \dots, t_n - n\Delta t\}$$

即 $D' \leq D$

证毕。

定理 2 对任意一个分配方案, 都可找到一个正规分配方案不差于该方案。

证明 对一个非正规分配方案, 假设数据传输启动的时刻记为 0, 根据播放顺序, 记各数据块的传输完成时刻为 $t_0, t_1, t_2, \dots, t_n$, 将该数列按由小到大顺序重新排列得到一个新的序列 $t_{i_0}, t_{i_1}, t_{i_2}, \dots, t_{i_n}$ 。序列 $i_0, i_1, i_2, \dots, i_n$ 为 $0, 1, 2, \dots, n$ 的一个排列。

设 i_k 为该序列中第一个数值与其下标不等的数, 即 $i_k \neq k$, 而当 $j < k$ 时有 $i_j = j$ 。因此, 必存在 m 满足 $k < m \leq n$, 使得 $i_m = i_k - 1$, 即 i_m, i_k 为两个连续块, 且 $t_{i_m} > t_{i_k}$, 交换块 i_m 与 i_k 的分配, 由引理 2 知得到的新分配方案不差于原来的分配方案。且序列的逆序数至少减小 1。

重复上述操作, 最多重复 $\frac{n(n+1)}{2}$ 次便可得到一个正规分

配方案, 且该分配方案的最小缓冲延迟不大于原来的分配方案。

证毕。

推论 如果一个正规分配方案是所有正规分配方案中缓冲延迟最小的分配方案, 则它必定是一个最优分配方案。

定理 3 由 ODTA_{P2P} 算法得到的分配方案是最优正规分配方案。

证明 不难看出, 如果将所有数据块都分配给供应节点 P_1 , 则各数据块传输完成的时间序列 N_1 为:

$$L/B_1, 2L/B_1, \dots, i \cdot L/B_1, \dots, n \cdot L/B_1$$

如果将所有数据块都分配给供应节点 P_2 , 则各数据块传输完成的时间序列 N_2 为:

$$L/B_2, 2L/B_2, \dots, i \cdot L/B_2, \dots, n \cdot L/B_2$$

如果将所有数据块都分配给供应节点 P_k , 则各数据块传输完成的时间序列 N_k 为:

$$L/B_k, 2L/B_k, \dots, i \cdot L/B_k, \dots, n \cdot L/B_k$$

...

如果将所有数据块都分配给供应节点 P_m , 则各数据块传输完成的时间序列 N_m 为:

$$L/B_m, 2L/B_m, \dots, i \cdot L/B_m, \dots, n \cdot L/B_m$$

...

将数列 N_1, N_2, \dots, N_m 合并并按由小到大顺序重新排列, 得到一个新的数列 N' 。截取 N' 的前 n 项组成一个新的数列 N'' , 则 N'' 为 N' 所有长度为 n 的子列中元素最小的子列。

由 ODTA_{P2P} 算法的分配方法知, 其所得到的分配方案所对应的各数据块的传输完成时间序列恰为 N'' , 因此该分配方案为正规分配方案。

下面证明由 ODTA_{P2P} 算法得到的分配方案的各数据块下载完成时间为所有正规分配方案中最早的。

对任意一个正规分配, 它的各数据块下载完成时间序列可

由下述方法得到:

(1) 对供应节点 P_k , 根据分配给它的数据块数 s , 取数列 N_k 的前 s 项, 得到数列 $N'_k, k=1, 2, \dots, m$;

(2) 将数列 N'_1, N'_2, \dots, N'_m 合并为一个数列, 并将其元素按从小到大顺序重新排列。

显然这样得到的一个数列为 N' 的一个子列, 且其各项不大于 N'' 的相应项。

因此, 由 ODTA_{P2P} 算法得到的分配方案的各数据块下载完成时间为所有正规分配方案中最早的, 由引理 1, 其缓冲延迟必为所有正规分配方案最小的, 再由定理 2 的推论知, 该方案为最优分配方案。

证毕。

3 算法的性能分析

ODTA_{P2P} 算法的算法复杂度为 $O(nm)$, 与 OTS_{P2P} 算法相同, 但 OTS_{P2P} 算法只有在各供应节点的上行带宽能表示为 $R_0/2^k$ 且总和恰好为 R_0 时 (R_0 为流媒体播放速率) 才能求解并得到最优解, 但 ODTA_{P2P} 算法能针对一般情况计算出最优分配结果。大量的实验验证了算法的有效性, 在这里不再多述。下面的例子来自于文献[3], 图 1 和图 2 分别给出了 OTS_{P2P} 算法和 ODTA_{P2P} 算法的分配结果, 它们的分配结果虽然不同, 但最小缓冲延迟相同, 均为 $4L/R_0$ 。

1	3	4	8
2		7	
6			
5			

图 1 OTS_{P2P} 算法的分配结果

1	2	4	5
3		6	
7			
8			

图 2 ODTA_{P2P} 算法的分配结果

例 流媒体文件被分为等长的 8 块, 4 个供应节点的上行带宽分别为 $R_0/2, R_0/4, R_0/8, R_0/8$, R_0 为流媒体的播放速率。

4 总结

在多对一的流媒体数据传输中, 如何在多个源节点间分配数据的传输任务, 是 P2P 流媒体系统设计的一个核心问题, 本文针对 CBR 流媒体的数据传输任务, 在保证流媒体连续播放条件下, 给出了一个以缓冲延迟为优化目标的最优化分配算法 ODTA_{P2P}, 并对算法的最优性给出了数学证明。通过实例验证了算法的有效性, 且其在计算复杂度与同类算法相同的前提下适合在无特定假设环境下应用。(收稿日期: 2007 年 7 月)

参考文献:

- [1] Tran D A, Hua K A, Do T T A peer-to-peer architecture for media streaming[J]. IEEE Journal on Selected Areas in Communications, 2004, 22(1): 121-133.
- [2] Xiang Z, Zhang Q, Zhu W, et al. Peer-to-peer based multimedia distribution service[J]. IEEE Transaction on Multimedia, 2004, 6(4): 343-355.
- [3] Xu D, Hefeeda M, Hambrusch S, et al. On peer-to-peer media streaming[J]. Journal of Distributed Computing Systems, 2002, 8(3): 251-259.
- [4] 刘亚杰, 张鹤颖, 窦文华, 等. P2P 分层流媒体中数据分配算法[J]. 软件学报, 2006, 17(2): 325-332.