

采用节点优先级的对等网络流媒体 请求量分配算法

任浩^{1,2,3}, 王劲林^{1,2}, 尤佳莉^{1,2}

(1. 国家核电技术有限公司北京软件中心, 100029, 北京; 2. 中国科学院国家网络新媒体
工程技术研究中心, 100190, 北京; 3. 中国科学院声学研究所, 100190, 北京)

摘要: 在多源协同调度的对等网格(P2P)流媒体系统中,针对现有请求量分配算法不能根据源节点性能动态调整向各节点分配的请求量,导致源节点带宽利用率不高以及服务器负载重的问题,提出一种基于节点优先级的请求量分配算法.该算法根据源节点的历史调度评价和可用数据量计算源节点优先级,调度时根据优先级动态调整分配给各源节点的请求量,优先向历史调度评价好和可用数据量多的源节点分配请求.仿真实验与实际系统测试表明,所提算法可以充分利用源节点带宽资源进行数据传输,较传统算法降低服务器负载 17.3%以上,提高了系统可扩展性.

关键词: 对等网络;流媒体;多源协同调度;请求量分配

中图分类号: TP393 **文献标志码:** A **文章编号:** 0253-987X(2011)12-0010-06

Efficient Request Allocation Algorithm for Peer-to-Peer Streaming System

REN Hao^{1,2,3}, WANG Jinlin^{1,2}, YOU Jiali^{1,2}

(1. State Nuclear Power Software Development Center, Beijing 100029, China; 2. National Network New Media Engineering Research Center, Institute of Acoustics, Chinese Academy of Sciences, Beijing 100190, China;
3. Institute of Acoustics, Chinese Academy of Sciences, Beijing 100190, China)

Abstract: To solve the problem that the existing request allocation algorithm is unable to adjust request amount allocated to each sender according to sender performance in multi-sender based P2P streaming system, leading to an insufficient bandwidth utilization and a heavy server pressure, a peer priority based scheduling algorithm is proposed. The algorithm calculates peer priority based on the historical scheduling evaluation and data ownership. The data request allocated to each peer is adjusted dynamically according to the priority during scheduling, the senders with higher historical scheduling evaluation and more wanted data are preferred to allocate among the data requests. The simulation and experiment in real P2P streaming system show that the proposed algorithm enables to make full use of peer bandwidth resources to transmit media data and reduces server pressure by 17.3% approximately, and the system scalability is improved effectively.

Keywords: peer-to-peer; streaming; multi-sender scheduling; request allocation

随着宽带网络的迅速发展,流媒体技术成为研究热点.由于流媒体应用的数据量大、服务时间长,传统客户端服务器模式下服务器很容易成为系统瓶颈.随着对等网络(P2P)的兴起,由于其具有可扩展

收稿日期: 2011-04-21. 作者简介: 任浩(1983—),男,博士生;王劲林(通信作者),男,研究员,博士生导师. 基金项目: “十一·五”国家科技支撑计划资助项目(2008BAH28B04);国家自然科学基金青年基金资助项目(60903218E0208);国家高技术研究发展计划资助项目(2009AA01A339,2008AA01A317).

网络出版时间: 2011-10-25

网络出版地址: <http://www.cnki.net/kcms/detail/61.1069.T.20111025.0856.001.html>

性高、部署成本低等特点,使得基于 P2P 的流媒体技术得到广泛应用^[1].

由于当前网络上单个源节点的网络传输速率难以满足媒体文件播放的速率要求,因此需要实时调度多个源节点为请求节点提供服务,即根据请求节点的数据需求、各个源节点的可用数据与带宽等,分别向各个源节点请求不同数量、不同编号的数据块,称之为多源协同调度.多源协同调度要在各源节点间合理分配数据请求量,使得源节点的服务能力得到充分利用以降低服务器负载,加速媒体数据在网络中的传播.

本文基于节点历史调度评价和拥有所需数据的情况提出源节点优先级模型,调度时根据优先级动态调整向各源节点的数据请求量,优先向高优先级的节点请求数据.该算法可以充分利用源节点的带宽资源进行数据传输,加速数据在网络中的传播.

1 相关技术

随机请求量分配算法^[2]对每个要请求的数据块随机分配给某个邻居节点,该算法尽管能够在各个邻居节点间取得一定的负载均衡,但不能根据邻居节点性能调整请求量分配,节点带宽利用率不高.

文献[3]提出了 OTS_{P2P} 的请求量分配算法.在该算法中,根据节点输出带宽的不同,将节点分成不同的类,第 n 类节点的输出带宽是 $R_0/2^n$, R_0 是媒体流码率.请求节点运行 OTS_{P2P} 算法,对 m 个供应节点按照一定策略分配带宽,再根据给定的带宽分配数据块请求,其目标是使得流会话中接收节点的缓冲延迟最小.但是,它对每类节点提供的输出速率进行了假设,要求每类节点的输出带宽是 $R_0/2^n$,而且要求被请求的数据块数目是 2^n ,缺乏一般性.而且,该算法是静态分配算法,不能根据网络环境变化进行调整,当系统出现拓扑变化和带宽波动等情况时,需要启动算法重新进行分配.

文献[4]提出了 MBDA_{P2P} 算法,它考虑了已分配数据块、待分配数据块以及各数据块产生的缓冲延迟,实现了请求量动态分配策略,能够适应不稳定的网络环境.但是,它只能在数据块提供节点不离开的情况下取得较好性能,而 P2P 网络的节点通常是不稳定的,数据提供节点有可能中途退出系统.

文献[5]从数据分发延迟的角度对服务器带宽消耗进行了分析,本文则针对现有研究不足,从最小化服务器带宽消耗角度对多源协同调度进行建模,并提出基于节点优先级的请求量分配算法(PBRA).

实验表明,PBRA 可以充分利用节点带宽资源进行数据传输,有效降低服务器负载.

2 多源协同数据调度模型

在 P2P 流媒体系统中,媒体文件被切割为等字节大小的块,节点以块为单位获取数据.图 1 为多源协同数据调度过程,请求节点 P_r 在任一调度周期内将其需要的 $|D_i|$ 个数据块分配给 m 个源节点.如何获得一个最优分配,可以充分利用各源节点的带宽资源,从而有效降低服务器负载,是多源协同调度的核心问题.

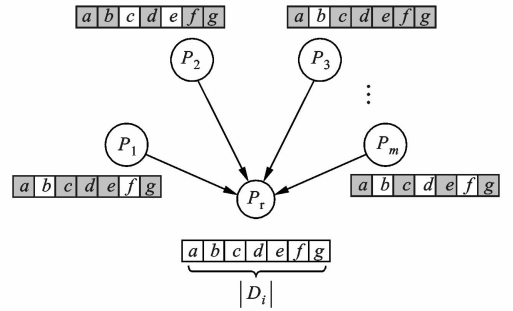


图 1 多源协同调度过程

设系统节点总数为 N ,数据调度周期为 τ , D_i 表示请求节点 P_i 在 t_c 时刻请求窗口内需要请求的数据片段集合, N_i^k 表示拥有 P_i 所需数据片段 k ($k \in D_i$) 的源节点集合, p_{ij}^k 表示节点 P_i 从源节点 P_j ($P_j \in N_i^k$) 在指定数据超时前成功请求到数据片段 k 的概率,用 I_i 、 O_i 表示节点 P_i 的下行和上行带宽.由于 P2P 流媒体系统中当服务器和普通源节点同时拥有所需数据时会优先向普通节点分配请求,因而若请求节点能够从普通源节点处及时获取更多所需数据,就可以有效降低服务器负载.因此,可以把优化目标转化为提高请求节点从邻居节点获取数据的成功率.设决策变量 x_{ij}^k 表示请求节点 P_i 是否将对片段 k 的请求分配给源节点 P_j ,于是优化目标可以表示为

$$\left. \begin{aligned} & \max \sum_{i \in N} \sum_{k \in D_i} \sum_{j \in N_i^k} p_{ij}^k x_{ij}^k \\ & (1) \sum_{j \in N_i^k} x_{ij}^k \leq 1, \forall k \in D_i \\ & (2) x_{ij}^k \in \{0, 1\}, \forall k \in D_i, j \in N_i^k \\ & (3) \sum_{k \in D_{ij}} \sum_{j \in N_i^k} x_{ij}^k \leq \tau I_i \\ & (4) \sum_{j \in N_i^k} \sum_{k \in D_i} x_{ij}^k \leq \tau O_j \end{aligned} \right\} \quad (1)$$

约束(1)保证每个数据片段 k 仅从至多一个拥有该片段的候选源节点请求,约束(2)表明多源协同调度是一个 0-1 规划,约束(3)、(4)表示多源协同调度时,请求的数据量不能超过请求节点的下行接收能力,也不能造成源节点服务过载. 式(1)的整数规划问题可以利用拉格朗日乘数法和次梯度法^[6]求解,但是算法复杂性较高,不适用于在实际系统中使用.

PBRA 中的 p_{ij}^k 与两个因素有关: P_j 有足够的带宽资源为 P_i 传输所请求的数据; P_j 拥有更多 P_i 没有的数据. 由此我们将基于源节点历史调度评价估计其可用带宽大小,再结合源节点拥有所需数据量多少定义节点优先级. 调度时根据优先级不同调整在各邻居节点间的请求量分配,通过充分利用节点带宽资源传输数据来提高节点请求成功率,从而降低服务器负载.

3 基于节点优先级的请求量分配算法

设节点 i 在 t_0 时刻加入系统,某次请求时刻为 t_c , j 为 i 本次请求的源节点之一. 节点优先级因子包括历史调度评价 $\bar{E}_{ij}$ 和内容价值度 $V_{i \rightarrow j}$.

3.1 历史调度评价

在每次调度开始时,请求节点都会检查上一调度周期内向各邻居节点调度出去数据的返回率. 若向某邻居节点 j 分配的数据请求全部成功返回,说明该邻居节点至少能够以与所分配请求量相对应的数据传输能力给自己提供服务. 根据累计成功返回数据的次数,说明按照这个能力向邻居节点 j 分配数据请求量比较稳定,可以尝试增大向 j 分配的请求量,以充分利用节点 j 的带宽资源;否则,若分配的数据请求没有及时成功获得,并累计达到一定次数,则确定向其分配的请求量超过该邻居节点的可用服务能力,那么在下一调度周期内将根据丢包情况酌情减少向该节点分配的请求量,以适应该节点的带宽能力.

为了保证请求节点能够流畅地播放视频,要求请求节点从各源节点获取数据的速率之和不低于视频码率. 因此,请求节点 i 对新加入源节点 j 的历史调度评价初值表示为

$$E_{ij}^0 = \frac{r\tau l^{-1}n^{-1}}{\tau} = r/l n \quad (2)$$

式中: r 为媒体流码率(B/s); τ 为调度周期(s); l 为数据块字节数(B); n 为请求节点的源节点个数. 式(2)表示将新加入源节点的历史调度评价初值定义

为一个调度周期内能够以码率 $1/n$ 的速率向请求节点传输数据.

确定源节点历史调度评价初值后,请求节点就会按照初值开始向各源节点分配数据请求,分配完毕后,在下一请求周期开始前会对源节点的请求服务情况进行评价. 设第 m 次调度请求节点 i 向 j 发送数据请求 q_m , 其中 q_m 表示为由请求数据块编号组成的量, $|q_m|$ 表示本次调度请求的数据块个数. 在下一调度周期开始前,请求节点会统计本次调度收到的请求数据块数目及收到这些数据消耗的时间,并根据统计结果计算本次请求的服务评价

$$E_{q_m}(i, j) = \sum_{\forall q_m} \frac{S(i, j)}{T(i, j)} \quad (3)$$

式中: $S(i, j)$ 表示在数据超时时刻前 i 收到 j 返回的数据片段数量,显然 $S(i, j) \leq |q_m|$, 且越接近 $|q_m|$ 表示返回的数据块数目越接近本次调度请求的数据块数目,源节点服务评价越高; $T(i, j)$ 表示节点 i 与 j 本次数据交互消耗的时间. 若累计多次 $E_{q_m}(i, j)$ 都较高,则请求节点会增大向对应源节点分配的请求量 q_m ; 反之则会减少向对应源节点分配的请求量 q_m . 于是, i 在 t_c 时刻对 j 的历史调度评价为

$$\bar{E}_{ij} = \frac{\sum_{m=1}^{(t_c - t_0)\tau^{-1}} E_{q_m}(i, j)}{(t_c - t_0)\tau^{-1}} \quad (4)$$

$\bar{E}_{ij}$ 可以反映出源节点 t_c 时刻可用上行带宽资源的相对大小,克服了传统方法需要精确测量节点物理带宽的弊端. 显然,带宽资源充足的源节点可以在更短时间内返回更多的数据块,即评价值 $\bar{E}_{ij}$ 越大. 向评价高的节点分配更多数据请求,能充分利用这些能力强的节点为请求节点传输数据,可以有效避免由于多次数据请求失败导致向服务器紧急请求情况的发生,有利于服务器负载的降低.

3.2 内容价值度

设系统中共有 m 部等字节大小的影片,影片编号依次为 $1, 2, \dots, m$. 每部影片又被切割为 k 个等字节大小的块,按照播放时间先后顺序对每个块依次编号为 $1, 2, \dots, k$. 于是有数据分割矩阵 $F = [f_{ij}]_{m \times k}$, 其中 f_{ij} 表示编号 i ($1 \leq i \leq m$) 的影片中编号为 j ($1 \leq j \leq k$) 的数据块.

设 N 为系统节点总数,节点按照泊松过程到达,到达率为 λ ,离开率为 δ . 根据 $M/G/\infty$ 排队模型, t 时刻有

$$N = N(t) = \lambda t - \delta t \quad (5)$$

每个节点 p 维护一系数矩阵 $\mathbf{A}=[a_{ij}^{(p)}]_{m \times k}$,其中

$$a_{ij}^{(p)} = \begin{cases} 1, & \text{如果 } p \text{ 拥有 } f_{ij} \\ 0, & \text{其他} \end{cases} \quad (6)$$

进而定义节点 $A, B, \dots, X$ 之间片段融合度为

$$S_{AB \dots X} = \sum_{i=1}^m \sum_{j=1}^k (a_{ij}^{(A)} \parallel a_{ij}^{(B)} \dots \parallel a_{ij}^{(X)}) \quad (7)$$

其中 $\parallel$ 表示逻辑或运算. 定义节点 A 对节点 B 的内容价值度为

$$V_{A \rightarrow B} = S_{AB} - \sum_{i=1}^m \sum_{j=1}^k a_{ij}^{(B)} \quad (8)$$

式(8)的物理意义为:源节点价值度越高,表明拥有越多自己没有的数据,优先向价值度高的节点请求数据可以加速媒体数据在 P2P 网络中的扩散,从而降低服务器负载.

3.3 节点优先级计算

邻居节点的优先级由请求节点根据优先级因子计算. 设 $Q(i, j)$ 表示 t_c 时刻源节点 j 对于请求节点 i 的优先级

$$Q(i, j) = (\overline{E}_{ij})^2 V_{i \rightarrow j} \quad (9)$$

其中幂数项表示同等条件下节点历史调度评价占更大比重,以体现优先向能力强节点分配数据请求的思想. 由于所有数据块等大小,因此用可请求数据块数目表示节点优先级,请求节点将根据源节点优先级决定向每个源节点的请求量.

3.4 算法描述

考虑到流媒体播放的时效性,将节点数据缓冲区分为紧急区和非紧急区. 紧急区包括节点即将播放的数据,紧急区数据将优先请求. 紧急区之后就是非紧急区,包括节点自身播放需要但播放时间并不紧急的数据,同时将与其他节点共享这些数据.

PBRA 数据调度算法.

(1) 初始化本次数据请求集合 Request-Collection.

(2) 根据式(4)、式(8)、式(9),计算所有源节点的优先级,并按照优先级降序排序形成 Candidate-List.

(3) 将紧急区中还没有得到的数据片段编号放入集合 Request-Collection.

(4) 计算非紧急区每个数据块的持有节点数目.

(5) 将非紧急区数据块编号按持有节点数目从少到多的顺序放入 Request-Collection.

(6) 对 Request-Collection 中顺序取出的每个数据块编号,从 Candidate-List 中找出当前优先级最

高的源节点,并将该编号放入对应源节点的请求队列 Request-Queue.

(7) 降低对应源节点优先级一个单位.

(8) 重复执行第 6、第 7 步直到 Request-Collection 中的所有片段都被分配.

(9) 向各个源节点发送分配的数据请求.

图 2 为 PBRA 的一个调度场景. 设本次调度请求节点 P 将请求 1~10 号数据块,其有 4 个源节点 P_1, P_2, P_3 以及 P_4 ,连线上的数字表示根据式(9)计算得到的优先级,则根据 PBRA 算法得到的各源节点请求量分配结果如表 2 所示.

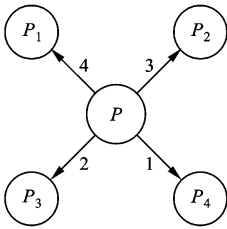


图 2 PBRA 的调度场景

表 2 PBRA 算法得到的源节点请求量分配关系

数据编号	1	2	3	4	5	6	7	8	9	10
源节点	P_1	P_1	P_2	P_1	P_2	P_3	P_1	P_2	P_3	P_4

可以看到,根据节点优先级分配数据请求量的算法能够充分利用各源节点的带宽资源进行数据传输,有利于服务器负载的降低.

3.5 算法时间复杂度分析

设节点缓冲区队列长度为 n 个数据包,其中紧急区长度为 $n/3$,非紧急区长度为 $2n/3$. 源节点数目为 m ,本次请求紧急区数据片段个数为 x .

算法第 1 步复杂度为 $O(1)$,第 2 步采用快速排序,复杂度为 $O(m) + O(m \lg m)$,第 3 步复杂度为 $O(x)$,第 4 步复杂度为 $O(m2n/3)$,第 5 步复杂度为 $O(2n/3)$,第 6~8 步复杂度为 $O(m2n/3)$,第 9 步复杂度为 $O(m)$. 因而 PBRA 总时间复杂度为

$$T = O(1) + O(m) + O(m \lg m) + O(x) + O(m2n/3) + O(2n/3) + O(m2n/3) + O(m) \quad (10)$$

一般情况下 $m < n$,所以式(10)可以简化为

$$T = O(nm) \quad (11)$$

另外,源节点历史调度评价可以在每次调度开始前利用节点空闲资源计算,在分配请求量时直接使用评估结果,因此有效减低了调度时的算法复杂度.

4 实验与分析

本节通过 P2PStrmSim 仿真器^[7]和实际系统运行验证算法性能。P2PStrmSim 是基于事件驱动的数据包水平 P2P 流媒体数据调度专用仿真器,该仿真器摒弃了一些对 P2P 网络流媒体协议不重要的因素(例如路由器中数据包的排队),提高了仿真效率,使仿真规模达到上万节点。采用康奈尔大学的网络延迟数据集^[8]模拟节点之间的网络延迟,这是实际互联网网络延迟的测量结果,是一个节点到节点之间的延迟矩阵($2\,500\times 2\,500$)。仿真中将任意一个节点对之间的延迟随机映射到该矩阵的某一个节点对上。

设置数据源服务器上行接入带宽为 10 Mb/s,媒体流码率为 300 kb/s。仿真总节点数为 10 000 个,节点以强度 $\lambda=0.25$ 的泊松分布加入系统,且在线时长按照 GridMedia 系统^[9]的 Trace 文件进行设置。为了模拟节点异构性,根据实际因特网测量结果^[10],将用户节点上行接入带宽聚类为 4 类,并规定各自所占比例,如表 3 所示。

表 3 节点接入带宽及比例

类型	上行带宽/ $\text{kb}\cdot\text{s}^{-1}$	下行带宽/ $\text{kb}\cdot\text{s}^{-1}$	比例/%
1	128	512	25
2	256	1 536	35
3	512	512	30
4	2 048	2 048	10

仿真中将 PBRA、OTS_{P2P} 以及 MBDA_{P2P} 请求量分配算法进行系统性能对比。

4.1 系统吞吐率

定义系统吞吐率为节点单位时间接收的有效数据包个数,反映了系统中节点的平均播放流畅程度,该参数越大表明播放越流畅,最大为媒体流码率。

图 3 比较了不同系统规模下不同请求量分配算法的吞吐率,可以看到 PBRA 的吞吐率在不同节点规模下均高于其他分配算法,且接近媒体流码率,表

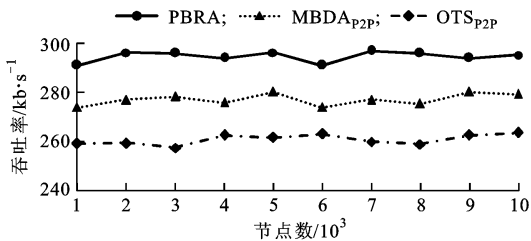


图 3 3 种算法的系统吞吐率比较

明大部分节点可以基本流畅地播放视频。由于 PBRA 算法根据源节点的历史调度情况动态调整请求量分配,能够有效适应网络动态性和节点可用服务能力的变化,提高了每个调度周期数据请求的成功率,使节点在播放截止时间前不能成功获取到数据块的比例降低,从而使系统吞吐率维持在较高水平。

4.2 启动延迟

在 P2P 流媒体系统中,为了避免由于网络抖动造成播放中断,每个节点会维护一个播放缓冲区,当播放缓冲区中数据片段达到一定数量时才启动视频播放。定义播放启动延迟为节点加入系统到开始播放视频的时间间隔,显然播放延迟越小用户体验越好。仿真中设节点缓存 3 Mb 的数据后开始播放。

图 4 描述了不同请求量分配算法的播放延迟比较,可以看到 PBRA 的播放延迟比其他算法都要小,90%的节点播放延迟在 12 s 以内。由于 PBRA 根据节点优先级确定向每个邻居节点分配的数据请求量,会向历史调度评价高和拥有更多所需数据的节点请求更多的数据,这样可以保证数据请求成功率,同时充分利用节点能力,在更短的时间内获得尽量多的数据,尽快启动视频播放,提供更好的用户体验。

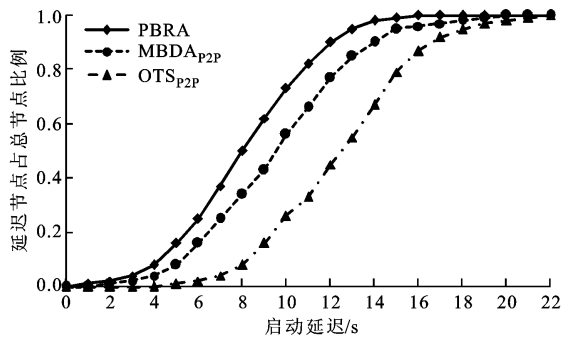


图 4 3 种算法播放延迟的比较

4.3 控制开销

P2P 流媒体系统中除了传输数据片段外还需要传输缓存映像(BM)报文、数据请求报文、连接建立与维护报文等控制报文。定义控制报文比率(CSPR)为控制报文传输速率与视频码率的比值。该参数用来衡量系统控制开销,显然控制报文比率越低系统控制开销越小,可扩展性越高。

图 5 比较了不同系统规模下不同请求量分配算法的控制开销,可以看到不同节点规模下 PBRA 的控制开销都低于其他分配算法。当系统节点规模达到 10 000 时控制报文比率也仅为 13% 左右,而 OTS_{P2P} 方法约为 19.7%,MBDA_{P2P} 约为 16%。由于 PBRA 向每个邻居节点请求的数据块个数与其服务

能力相当,提高了每次调度的数据请求成功率,从而降低了由于数据超时进行重传请求的概率,减少了不必要的请求包数目,有利于控制开销的降低。

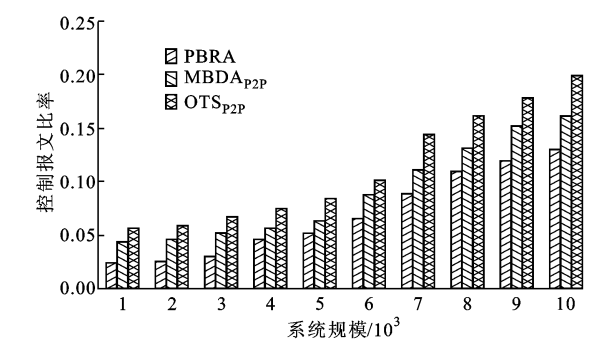


图 5 3 种算法的控制开销比较

4.4 媒体源服务器负载

将 PBRA 应用到 P2P 流媒体系统 NOVA^[11]中,观察服务器负载的优化情况。NOVA 是由国家网络新媒体工程技术研究中心研发的 P2P 流媒体系统,可以同时支持直播和点播两种业务。实验中系统运行在 LAN 网络环境,节点上行物理带宽均为 1 Mb/s。为实现节点异构性,利用 Bandwidth Controller Manager 带宽限制软件,根据表 3 对节点可用上行带宽进行设置。运行节点总数为 100,其他参数与仿真一致。

定义归一化服务器带宽消耗(NCSB)为服务器实际出流量速率与最小服务器带宽需求的比值。最小服务器带宽需求为

$$M_{\text{SBN}} = \max\left\{r, Nr - \sum_{i=1}^{M_{\text{SBN}}} u_i\right\} \quad (12)$$

式中: r 为视频码率; N 为系统中节点数; u_i 为每个节点的上行带宽。式(12)的物理意义为:当节点的上行带宽之和足够大时, M_{SBN} 为一倍的媒体流速率,即媒体服务器只需要为每一个数据包推送一份拷贝到 P2P 网络中;当节点总上行带宽小于最小总带宽需求(Nr)时,则服务器需要将剩余所需带宽补齐。并且,NCSB 总是大于 1,且越接近 1 表示服务器负载越小。

图 6 比较了不同系统规模下不同请求量分配算法的服务器带宽消耗,可以看到 PBRA 的服务器带宽消耗在不同节点规模下均低于其他分配算法,比 OTSP_{2P} 平均降低约 25%,比 MBDA_{P2P} 平均降低约 17.3%。图 6 反映了 P2P 流媒体的重要特征,即随着系统规模的扩大服务器负载没有显著增大,相反由于网络中有更多节点共享自己的上行带宽资源,服务器负载得到减轻。由于 PBRA 算法根据源节点

历史调度情况和拥有所需数据多少动态调整在各源节点间的请求量分配,提高了源节点上行带宽利用率,可以更充分地利用能力强的源节点带宽资源为请求节点传送更多所需数据,所以同等系统规模下服务器负载更小。

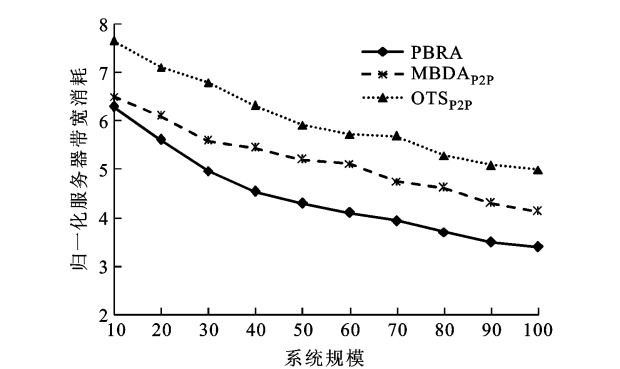


图 6 服务器带宽消耗比较

5 结 论

本文研究 P2P 流媒体系统中多源协同调度的请求量分配问题,在将多源协同调度建模为最优化问题的基础上引入节点优先级模型,提出一种基于节点优先级的请求量分配算法 PBRA。PBRA 根据历史调度评估值和拥有可用数据量在各源节点间分配数据请求量,优先向历史调度评价好以及拥有更多所需数据的源节点分配数据请求,以实现向能力强的节点请求更多数据,从而充分利用源节点的带宽资源进行数据传输。仿真实验和实际系统运行表明,本算法提高了普通节点带宽资源利用率,有效降低了媒体服务器负载。在下一步工作中将就如何优化服务器带宽消耗进行进一步研究,以提高系统可扩展性。

参考文献:

[1] 黄泳翔,钱德沛,伍卫国,等. 可分级视频编码覆盖网非一致性随机成员构建算法[J]. 西安交通大学学报,2009,43(6):1-4.
HUANG Yongxiang, QIAN Depei, WU Weiguo, et al. Non-uniform random membership management to construct overlays for transferring scalable video coding[J]. Journal of Xi'an Jiaotong University, 2009, 43(6):1-4.

[2] PAI V, KUMAR K, TAMILMANI K, et al. Chain-saw:eliminating trees from overlay multicast [M]// Lecture Notes in Computer Science; 3640. Berlin, Germany: SpringerLink, 2005:127-140.

[3] XU Dongyan, HEFEEDA M, HAMBRUSCH S, et

- al. On peer-to-peer media streaming[EB/OL]. [2011-03-22]. [http:// friends. cs. purdue. edu/pubs/IC-DCS02. pdf](http://friends.cs.purdue.edu/pubs/IC-DCS02.pdf).
- [4] 杨薇薇,黄年松. 一种 P2P 流媒体数据传输任务分派算法[J]. 华中科技大学学报: 自然科学版, 2005, 33(5):3-6.
- YANG Weiwei, HUANG Niansong. Data assignment of P2P media streaming[J]. Journal of Huazhong University of Science and Technology, 2005, 33(5):3-6.
- [5] LU Yifeng, REN Hao, WANG Jinlin. Real-time performance vs. server bandwidth cost in peer-to-peer streaming system [C]//Proceedings of the International Conference on Computer and Electrical Engineering, Piscataway, NJ, USA: IEEE, 2008: 286-290.
- [6] PALOMAR D, CHIANG M. Alternative distributed algorithms for network utility maximization: framework and applications[J]. IEEE Transactions on Automatic Control, 2007, 52(12):2254-2269.
- [7] ZHANG Meng, ZHANG Qian, SUN Lifeng, et al. Understanding the power of pull-based streaming protocol: can we do better? [J]. IEEE Journal on Selected Areas in Communications, 2007, 25(9):1678-1694.
- [8] Cornell University. Meridian project [EB/OL]. [2011-02-10]. [http:// www. Cs. Cornell. Edu/people/egs/meridian/](http://www.Cs.Cornell.Edu/people/egs/meridian/).
- [9] ZHANG Meng, XIONG Yongqiang, ZHANG Qian, et al. Optimizing the throughput of data-driven peer-to-peer streaming [J]. IEEE Transactions on Parallel and Distributed Systems, 2010, 20(1):97-110.
- [10] CIULLO D, GARCIA M A, HORVATH A, et al. Network awareness of P2P live streaming applications: a measurement study [J]. IEEE Transactions on Multimedia, 2010, 12(1):54-63.
- [11] LI X. NOVA[EB/OL]. [2010-11-30]. <http://dspnova.8800.org/>.

(编辑 杜秀杰)