

文章编号:1001-9081(2009)10-2620-04

一种基于激励机制的 P2P 流媒体数据调度模型

张秋余, 邓涛, 袁占亭

(兰州理工大学 计算机与通信学院, 兰州 730050)

(dengtao27@mail2.lut.cn)

摘要:针对 P2P 流媒体中现有算法在数据调度时不够公平的问题,提出了一种新的具有激励作用的算法模型。该算法模型可以根据节点的丢包率、贡献效率动态调节节点带宽分配以达到接收节点利益理性最大化和系统整体公平,提高了系统传输效率。仿真实验表明,该算法模型促进了节点的参与传输调度,达到了维持系统公平的目的。

关键词:对等网络;流媒体;激励机制;数据调度;贡献效率

中图分类号: TP393 **文献标志码:** A

Data scheduling model based on incentive for P2P media streaming

ZHANG Qiu-yu, DENG Tao, YUAN Zhan-ting

(School of Computer and Communication, Lanzhou University of Technology, Lanzhou Gansu 730050, China)

Abstract: In order to solve the unfairness in the existing algorithm on data scheduling in P2P media streaming, a new algorithm model having incentive function was proposed. It can dynamically adjust bandwidth allocation of peers according to their packet loss and contribution rate to realize maximum of receiving peer's interests rationally and fairness of the whole system, and enhance system transmission efficiency. The simulation results show that this algorithm model motivates peers to participate in data scheduling and maintains system fairness.

Key words: Peer-to-Peer (P2P); streaming media; incentive; data scheduling; contribution rate

0 引言

随着互联网的普及与飞速发展,越来越多的用户正在享受和期待高质量的在线媒体服务。P2P 流媒体技术已成为研究的热点。在多源 P2P 流媒体系统中,各发送节点的上传带宽、丢包率等因素差异较大,涉及因素较多,对多源节点进行有效合理的数据调度仍是一个复杂的问题^[1-2]。

现有关于多源节点数据调度的研究已取得不少的成绩^[3-7]。在多个提供不同发送速率的发送节点向一个接收节点发送数据的研究中,其主要工作集中在如何分配多个供给节点传输带宽上。文献[5]作者提出以最小缓冲延迟为目标的数据分配算法 OTS,而文献[6]作者以丢包率最小为目标提出了多发送端的速率分配算法。对此国内也有相关的研究,它们共同的特点更偏于从接收节点利益最大化的角度进行考虑。近些年来学者针对 P2P 系统中“搭便车”等现象致力于开展激励机制的工作^[8-12],但激励机制主要作为 P2P 系统中一个单一领域来研究。

本文拟将激励机制中的激励思想运用到比较现实的数据调度算法中,提出了一个新的具有激励作用的多源 P2P 节点数据调度模型。此模型不仅可实现接收节点动态调整接收媒体数据的调度方案,而且可以对各发送节点起到激励上传作用,因此既满足单个节点利益的理性最大化,也使整个节点调度系统中的各类节点都分时做出贡献以达到系统公平性和激励的平衡。

1 数据调度模型的建立

在实际的 P2P 数据传输系统中,丢包率是一个不容忽视

的因素,它直接影响着接收节点有效接收的数据量的大小和效率。文献[6]提出的以丢包率最小为目标的数据分配算法是一个较为实际可行的模型。其算法思想是:根据 TFRC^[13]计算出第 i 个发送节点 t 时刻的 TCP 友好带宽 $B(i, t)$,即可用上限带宽 $L(i, t)$, $S(i, t)$ 分别为发送节点 i 在时间间隔 $(t, t + \Delta t)$ 的丢包率和发送速率, $S_{req}(t)$ 是接收节点媒体流正常播放所需的速率。确定 $S(i, t)$ 大小使 $F(t)$ 最小:

$$F(t) = \sum_{i=1}^n L(i, t) S(i, t)$$
$$\text{s. t. } 0 \leq S(i, t) \leq B(i, t), \sum_{i=1}^n S(i, t) = S_{req}(t) \quad (1)$$

即求 $\underset{S(i, t)}{\operatorname{argmin}} \left[\sum_{i=1}^n L(i, t) S(i, t) \right]$

在此算法中最关键的部分是将发送节点的丢包率进行由小到大的排序,对丢包率最小的发送节点 i 分配该发送节点的上限带宽 $B(i, t)$,然后对丢包率次小的发送节点 j 分配该发送节点的上限带宽 $B(j, t)$,以后以此类推,直至发送节点的上限带宽之和等于或超过媒体流回放速率。此算法使得接收节点最大化利用了发送节点的有效传输带宽,使尽可能少的高效传输节点优先列入传输队列,而诸如此类传输调度算法大都以接收节点利益最大为目标。传统激励机制更侧重于抑制“搭便车”行为,采取下载文件消耗收益,上传文件做出贡献,节点贡献愈多得到的资源分配愈多。

以上算法接收节点仅以丢包率最小为目标,通过分配传输带宽使得自身利益最大。这在客观上必然会造成少数传输效率高的节点一直参与传输,负担沉重,影响其参与传输的积极性,同时也造成了系统内的其他节点不能有效参与传输,很

收稿日期:2009-04-07;修回日期:2009-06-02。 基金项目:甘肃省自然科学基金资助项目(0803RJZA024)。

作者简介:张秋余(1966-),男,河北辛集人,副研究员,主要研究方向:软件工程、网络计算、多媒体通信技术; 邓涛(1984-),男,陕西西安人,硕士研究生,主要研究方向:计算机网络与通信; 袁占亭(1961-),男,陕西扶风人,教授,博士生导师,主要研究方向:计算机网络与软件、通信与信息系统、控制理论与控制工程。

少有机会享受收益,因此不能实现系统公平和整体利益最大。本文将激励机制中贡献思想运用至此数据调度算法中,既可以扩大参与传输的节点数目而且可以使各类节点分时分批加入传输,以弥补上述缺点,还能使系统传输效率得到提高。近些年来随着科学技术和设备飞速发展,各节点的仪器水平均有不同程度的提高,也创造了节点积极参与 P2P 传输的优越条件。对原有算法模型进行改进,给每个发送节点加入贡献因子: $\alpha(i,t)$,它表示 i 节点从初始加入系统到 t 时刻的贡献量 $C(i,t)$ 与所有节点贡献量之和之比。

i 节点的贡献量 $C(i,t)$ 为 i 节点上传量减去 i 节点从系统中其他节点处下载量,即为:

$$C(i,t) = \int_0^t (U_i - D_i) dt$$

在实际中, i 节点的上传量也可表示为其他节点从 i 节点所获得的下载量,因此上式变为:

$$C(i,t) = \int_0^t \left(\sum_{j=1}^N D_j - D_i \right) dt$$

那么贡献因子为:

$$\alpha(i,t) = \frac{\int_0^t (U_i - D_i) dt}{\sum_{i=1}^N \int_0^t (U_i - D_i) dt}$$

此外,节点的在线时长直接影响 P2P 流媒体系统数据传输稳定性,一个频繁进出系统的节点在传输会话中传输性能波动较为剧烈。因此无论考虑接收节点还是系统整体稳定利益,都应在数据调度中尽可能避免选择此类忽进忽出的“不忠诚节点”,而节点的各在线时长段愈小、愈多,愈不利于系统。因此,在线时长因素可取为节点加入系统后各时间段平方和。这样,既可考虑到了绝对时长大小也考虑到了累加时间段的优化效果。例如,两个节点累计在线时长都为 10 个时间单位,均进出系统两次,时长段时间单位分别为 1 与 9 的节点就比 5 与 5 的节点效果好($1^2 + 9^2 > 5^2 + 5^2$)。这样将单个节点在线时长的因素也考虑进此模型中,即为:

$$\sum_{i=1}^n (t_e - t_s)_i^2$$

其中 t_s 为节点加入系统时刻, t_e 为该节点退出系统的时刻, i 指第 i 次进出系统。这样,此模型就可以演变为:

$$F(t) = \sum_{i=1}^n \frac{S(i,t)L(i,t)\alpha(i,t)}{\sum_{j=1}^m (t_e - t_s)_j^2}$$

$$\text{s. t. } 0 \leq S(i,t) \leq B(i,t), \sum_{i=1}^N S(i,t) = S_{req}(t)$$

即求:

$$\operatorname{argmin}_{S(i,t)} \sum_{i=1}^n \frac{S(i,t)L(i,t)\alpha(i,t)}{\sum_{j=1}^m (t_e - t_s)_j^2}$$

$L(i,t)$ 为发送节点 i 在 t 时刻的丢包率, $\alpha(i,t)$ 为发送节点 i 在 t 时刻的贡献因子, $\sum_{j=1}^m (t_e - t_s)_j^2$ 为发送节点 i 累计 j 段在线时长的平方和。

2 数据调度模型的改进与分析

这样在此算法模型中丢包率 $L(i,t)$ 、贡献因子 $\alpha(i,t)$ 、在线时长三个因素相互制衡。直觉上,按此模型两个节点在数据调度中大致会产生以下三种情况:当两节点贡献因子和在线时长两个因素综合大致相等时,丢包率较小的节点会被选入

传输会话,这样更有利于接收节点的接收效率;当两节点丢包率和在线时长两个因素综合大致相等时,贡献因子小的节点会被选入传输会话,这样更有利于系统的激励作用;当两节点丢包率、贡献因子两个因素综合大致相等时,在线时长大的节点会被选入传输会话,这样更利于系统传输的稳定性。其实可以将 $\frac{\alpha(i,t)}{\sum_{j=1}^m (t_e - t_s)_j^2}$ 看作单位时间内的贡献因子,即贡献效率。

但是在线时长和贡献因子这两因素可能会产生一定的矛盾:贡献因子小的节点在线时长可能也小,所以可能不被选上;而贡献因子大的节点在线时长可能也大,所以可能被选上。为了在具体的系统中使三种因素影响程度效果不同,不妨给丢包率 $L(i,t)$ 、贡献因子 $\alpha(i,t)$ 、在线时长 $\sum_{j=1}^m (t_e - t_s)_j^2$ 加以特定的影响系数 α, β, γ , 那么改进后的模型即为:

$$F(t) = \sum_{i=1}^n \frac{S(i,t)L(i,t)\alpha(i,t)^\beta}{\left(\sum_{j=1}^m (t_e - t_s)_j^2 \right)^\gamma}$$

$$\text{s. t. } 0 \leq S(i,t) \leq B(i,t), \sum_{i=1}^N S(i,t) = S_{req}(t)$$

这样按照改进后模型对各节点分配传输带宽将更为合理。首先被选中的节点一定是丢包率最小、贡献效率最小的节点,分配给它的可用上限带宽;接着是丢包率次小或者贡献效率次小的节点,也分配给它的可用上限带宽;接着所选的节点丢包率或贡献效率依次增加,依此类推直至发送节点上限带宽之和等于或超过媒体流回放速率,以后的节点不用分配带宽。这样容易被选中可能是丢包率小的,也有可能是贡献效率较低的,如果是前者有利于接收节点最优利用发送节点的下载资源,如果是后者有利于整个激励机制在系统中发挥作用从而达到系统的公平性。

事实上,贡献效率较低的节点与贡献效率高的节点、丢包率较低的节点与丢包率高的节点都处于一种动态平衡之中,而它们的定量关系可由设定具体影响系数来确定。一个新加入此系统的节点,由于没有上传或上传量相对很小,因此贡献效率自然很小,这样在丢包率与其他节点相差不大时就被优先选入传输会话,而随着贡献效率的增大该节点自然会后移直至跳出会话。而当节点跳出传输会话后未做贡献,随时间增长贡献效率降低,因此一段时间后将再次被选入传输会话。再从丢包率的角度考虑,丢包率小的节点被选中几率大,做出贡献多,接收节点贡献因子也会随之增大,因此也会渐渐后移跳出会话。这样节点不论贡献效率的高低,都根据其模型值的变化而分时加入传输会话。

3 实例演示及算法

下面举一个实例说明传输调度的过程,为了避免过于复杂暂不考虑在线时长因素,按此模型对系统内一系列节点进行排序, α, β 暂定取 1。假设有 8 个节点 $S_1 \sim S_8$ 在时刻 t 丢包率 $L = [0.01, 0.03, 0.05, 0.10, 0.15, 0.05, 0.07, 0.09]$, 贡献因子 $\alpha = [0.03, 0.05, 0.04, 0.05, 0.07, 0.09, 0.08, 0.06]$, 接收节点的媒体回放速率 $R = 1$ Mbps, 8 个节点的 TCP 友好带宽 $B = [150 \text{ Kbps}, 220 \text{ Kbps}, 140 \text{ Kbps}, 200 \text{ Kbps}, 220 \text{ Kbps}, 250 \text{ Kbps}, 280 \text{ Kbps}, 240 \text{ Kbps}]$, 那么 8 个节点 $S_1 \sim S_8$ 按照该模型排序 $S_1, S_2, S_3, S_6, S_4, S_8, S_7, S_5$ 。由于 $B_1 + B_2 + B_3 + B_6 + B_4 + B_8 = (150 + 220 + 140 + 250 + 200 + 240) \text{ Kbps} = 1200 \text{ Kbps} > 1024 \text{ Kbps} = 1 \text{ Mbps}$, 达到媒体回放速率 R , S_8 仅分配 64 Kbps,

而 S_7, S_5 未分配带宽, 即不参与此次传输会话。

数据调度带宽分配算法的伪代码如下:

```

Given a set of  $N$  peers in the System;
//  $N$  is the number of peers in P2P media streaming system
struct peer //structure the peer within its attribute value
{
    sendrate;
        // the sending rate that is allocated by the model algorithm
    loss; //packet loss rate
    bestrate; //optimal sending rate(TCP - Friendly Bandwidth)
    duration; //  $\sum_{i=1}^n (t_e - t_s)_i^2$ 
    contr; //contribution factor
    model; //record  $\frac{L(i,t)^\alpha \alpha(i,t)^\beta}{[\sum_{j=1}^m (t_e - t_s)_j^2]^\gamma}$ 
};
struct peer peerqueue[  $n$  ];
    Compute the loss rate  $L_i$ , contribution factor  $\alpha_i$ , duration  $t_i$ ,
    optimal sending rate  $B_i$  for each peer  $i$ ;
    Set  $\alpha, \beta, \gamma$  for  $F_i = \frac{L_i^\alpha * \alpha_i^\beta}{t_i^\gamma}$ ;
    //  $\frac{L_i^\alpha * \alpha_i^\beta}{t_i^\gamma}$  is short for  $F_i = \frac{L(i,t)^\alpha \alpha(i,t)^\beta}{[\sum_{j=1}^m (t_e - t_s)_j^2]^\gamma}$ 
    Compute the  $F_i$  for each peer  $i$ ;
    For (  $i = 0; i < n; i++$  )
    {
        peerqueue[  $i$  ]. sendrate = 0;
        peerqueue[  $i$  ]. loss =  $L_i$ ;
        peerqueue[  $i$  ]. bestrate =  $B_i$ ;
        peerqueue[  $i$  ]. contr =  $\alpha_i$ ;
        peerqueue[  $i$  ]. duration =  $t_i$ ;
        peerqueue[  $i$  ]. model =  $F_i$ ;
    }
struct peer A [  $n$  ];
// Sort  $F_i$  in ascending orders and store peerqueue[  $i$  ] in
array A [  $n$  ]
For(  $i = 0; i < n; i++$  )
{
    struct peer tmp = peerqueue[  $i$  ];
    For(int  $j = i + 1; j < n; j++$  )
    {
        If ( tmp. model > peerqueue[  $j$  ]. model)
            tmp  $\leftrightarrow$  peerqueue[  $j$  ];
    }
    A[  $i$  ] = tmp;
}
//For(  $i = 0; i < n; i++$  )
int  $i$ ;
//allocate bandwidth to peers with the largest N model
value//(datascheduling)
while (  $i = 0; sumB < R; i++$  )
{
    sumB = 0;
    A[  $i$  ]. sendrate = A[  $i$  ]. bestrate;
    sumB = sumB + A[  $i$  ]. sendrate;
}
A[  $i + 1$  ]. sendrate = R - sumB;
//allocate bandwidth to the last peer in scheduling conversation
Set the each peer's sending rate with peerqueue[  $i$  ];

```

4 性能测试与分析

为了更好地分析此算法模型的性能, 本系统采用仿真实验进行性能测试。其配置为: Intel Core2 Duo CPU T5450, 1 GB内存。操作系统 WindowsXP。实验使用 Omnet ++ 仿真软件, 加载以上算法模型。

算法模型仅考虑单个接收节点向多个发送节点获取数据的情况。实验中调度算法周期性地执行, 但考虑不增加网络负担将周期定为 10 s, 为突出贡献效率的影响 α 设为 0.75, β 、 γ 设为 1, 所使用的媒体流编码率为 500 Kbps。系统中共构造了 100 个节点(设定 70 个参加传输, 另 30 个不参与传输且与前者随机交换出入), 其中 30 个为活跃节点, 其贡献效率在 [0.06, 0.10] 波动; 30 个为惰性节点, 其贡献效率在 [0, 0.04] 波动; 40 个为其他节点, 其贡献效率在 [0.04, 0.06] 波动。节点的丢包率在 [0.01, 0.05] 均匀分布。系统在初始化时, 贡献效率即分配。

第一个实验中观测了 80 个单位时间段内三类节点在被选入传输会话中节点个数的分布情况, 这是为了验证该算法模型是否对三类节点起到了激励上传的作用。进一步, 为了评估此算法模型的系统传输效率, 在实验中加入对比方案, 该方案算法随机选取传输节点直至它们的上限带宽之和达到媒体流回放速率。

表 1 三类节点在传输调度中分布情况及系统传输效率对比

时刻/s	活跃节点数	惰性节点数	其他节点数	参与传输会话节点数	
				本算法	随机
10	10	6	4	20	25
20	8	7	6	21	17
30	6	10	7	23(0.022)	31(0.031)
40	3	12	7	22	39
50	5	10	5	20(0.019)	27(0.029)
60	5	9	5	19	42
70	5	9	7	21(0.020)	36(0.037)
80	4	9	6	19	13

从该实验可以看出在绝大多数时间内惰性节点占有比例最大, 其他节点次之, 活跃节点最小。由于贡献效率在实验中影响设定比较大, 说明此算法模型起到了激励公平的作用。再次, 随机选取参与传输会话的节点个数会出现不稳定的现象, 而多数情况下, 与实施算法模型参与传输会话节点的个数相比也较多, 在实验中抽取时刻 30、50、70 s 求其参与传输会话节点的平均丢包率(括弧中数据)对比, 后者均较大。说明算法模型在一定程度上选取了数量较少的传输效率较高即丢包率小的节点参与传输会话, 提高了系统传输效率, 符合实验的设计想法。

第二个实验中任选了 1 个活跃节点和 1 个惰性节点, 观测了它们 100 个单位时间内 5 次平均位序情况以对比活跃、惰性两类节点在调度算法中具体动态表现情况。

从图 1 可以看出活跃节点在实施算法模型后大多情况位于选中节点临界位序平滑线之上, 而惰性节点反之。活跃节点和惰性节点在开始时都在传输会话之中, 经过一段时间后由于贡献效率增大逐渐跳出传输会话, 随着贡献效率的降低再次趋于加入传输会话。而活跃节点比惰性节点较快跳出传输会话, 较慢跳回传输会话, 且跳出后离临界位序也较远, 这是由于在传输会话前期活跃节点较惰性节点贡献效率大所

致,贡献愈多,“休息”时间愈长,这符合激励公平的性质。

第三个实验中为了对比基于激励机制的数据调度算法和以丢包率最小为目标发送速率分配算法的性能,选取了上个实验中的活跃节点分别加载以上两种算法模型,观测两者的位序对比情况。

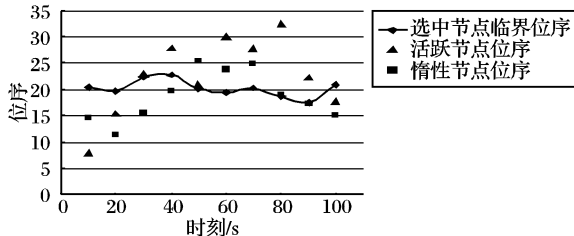


图 1 活跃节点与惰性节点位序对比

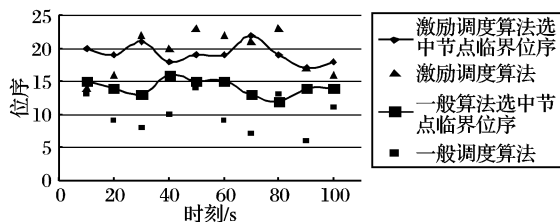


图 2 两种调度算法下节点位序对比

从对比中可以发现,激励调度算法与传统算法相比,选中更多的节点参与传输会话,因此节点临界位序更高,而选取的活跃节点分时跳出或跳入传输会话,达到了激励平衡和维持系统公平的作用,符合此算法模型的设计目的;而在传统算法模型中,此活跃节点长期处于传输会话中,未能达到上述目的。

5 结语

本文在总结以往各种 P2P 流媒体传输中数据调度的算法的基础上,提出了新的具有激励作用的多源节点数据调度模型。此算法模型考虑到了接收节点的以丢包率最小选择节点的利益,同时增加了贡献效率因素,通过对模型值的排序分配各发送节点的传输带宽,从而达到系统公平和激励平衡。此算法模型已在实验中初步验证,具有一定的参考意义,但仍需在具体环境中完善改进,例如接收节点需发送控制包调节传输带宽,这造成了一定的网络拥塞,在大量节点调度中可能会造成网络负担过重。另外,第二个实验效果不是特别明显,仍需改进。

(上接第 2619 页)

这主要是因为负载瞬间加大的行为造成的。随着 t_{unload} 的逐渐增加,这种瞬间行为的影响逐渐减弱,4 条曲线的覆盖率都相对有所提高。axp0 作为背景负载,在面对短任务时覆盖率是最低的,同时也证明了较重的主机负载对任务执行时间的影响较大;axp7 和 sahara 作为背景负载时,其覆盖率相当,都比较接近理想覆盖率,这也与图 3 所显示的预测精度结果相符合。

4 结语

仿真结果证明了在网络环境中利用主机负载来预测任务执行时间具有实际操作意义,预测效果较好。由任务执行时间预测的平均相对误差和覆盖率的仿真结果可知,背景负载过重,对任务执行时间预测精度影响较大,但随着 t_{unload} 的逐渐增加,不同背景负载的覆盖率都相对有所提高。因此,基于主机负载的任务执行预测算法在网络任务调度等工作中能满足实际应用要求。

参考文献:

- [1] 龚海刚,刘明,毛莺池,等. P2P 流媒体关键技术的研究进展[J]. 计算机研究与发展 2005,42(12): 2033-2040.
- [2] COLUBCHIK L, ABDOUNI B, LEE A W J. Video streaming over the Internet[EB/OL]. [2009-01-01]. http://imsc.usc.edu/research/project/videostream/videostream_nsf7.pdf.
- [3] KWON J B, YEOM H Y. Multimedia content distribution over peer-to-peer network[EB/OL]. [2009-02-01]. www.cse.msu.edu/icdcs/posters/final/01_s.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): 26-28.
- [5] TU Y C, LEI S. Towards cost-effective on-demand continuous media service: a peer-to-peer approach [EB/OL]. [2009-02-01]. www.cs.purdue.edu/research/technical_reports/2003/TR%2003-023.pdf.
- [6] XU D, HEFEEDA M, HAMBRUSCH S, et al. On peer-to-peer media streaming [C]// ICDCS'02: Proceedings of IEEE International Conference on Distributed Computing System. Washington, DC: IEEE Computer Society, 2002: 363-371.
- [7] NGUYEN T, ZAKHOR A. Distributed video streaming over Internet [C]// Proceedings of the SPIE Conference on Multimedia Computing and Networking. Philadelphia: SPIE, 2002: 186-195.
- [8] FELDMAN M, PAPADIMITRIOU C. Free-riding and whitewashing in peer to peer system[J]. IEEE Journal on Selected Areas in Communications, 2006, 24(5): 228-236.
- [9] 万健,郑若艇,徐向华. P2P 网络中激励机制研究[J]. 计算机应用, 2007, 27(9): 2202-2205.
- [10] 余一娇,金海. 对等网络中的搭便车行为分析与抑制综述[J]. 计算机学报, 2008, 31(1): 1-15.
- [11] 冯健,房鼎益,陈晓江. P2P 流媒体激励机制研究[J]. 计算机科学, 2008, 35(5): 29-31, 37.
- [12] 徐海涓,郑相全,齐守青,等. 一种新的基于博弈论的 P2P 激励机制[J]. 计算机应用研究, 2008, 25(9): 2787-2789.
- [13] FLOYD S, HANDLEY M, PADHYE J, et al. Equation-based congestion control for unicast applications[C]// SIGCOMM'00: Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication. New York: ACM Press, 2000: 43-56.

参考文献:

- [1] 都志辉,陈渝,刘鹏. 网格计算[M]. 北京:清华大学出版社, 2002.
- [2] REED D A, MENDEC C L. Intelligent monitoring for adaptation in grid applications[J]. Proceedings of the IEEE, 2005, 93(2): 426-435.
- [3] JAMSHED M, KHALIQUE S, SUGURI H, et al. Grid node monitoring architecture for autonomous resource management[J]. Broadband Networks, 2005, 2: 1362-1369.
- [4] DINDA P A, O'HALLARON D R. The statistical properties of host load[C]// LCR 98: Fourth Workshop on Languages, Compilers, and Run-time Systems for Scalable Computers. The Netherlands: IOS Press, 1998: 1-23.
- [5] Load trace archive[EB/OL]. [2009-01-01]. <http://www.cs.northwestern.edu/~pdinda/LoadTraces/>.
- [6] Load trace playback[EB/OL]. [2009-01-01]. <http://www.cs.northwestern.edu/~pdinda/LoadTraces/playload/playload>.