

P2P 系统中自助式负载均衡下载的实现

陶明亮¹, 张 曙², 王李军¹, 卜红建¹

(1. 中国科学技术大学软件学院, 合肥 230052; 2. 中国科学技术大学苏州研究生院, 苏州 215123)

摘要: 在资源分配和共享领域中 P2P 技术成为实现资源共享的有效应用方式。基于 P2P 技术, 资源的分配效率得到极大的提高, 但是由于网络带宽、资源位置等因素影响, 资源下载负荷往往是不均衡的。对此, 该文提出了基于网络带宽, 在多资源单用户和单资源多用户这两种情况下由资源提供者自我发现、自我请求其他有效资源的方式实现资源负载均衡分布。实验结果表明, 该方法是有用的。

关键词: P2P 下载; 资源共享; 自助式负载均衡

Implementation of Self-load-balancing Download in P2P System

TAO Ming-liang¹, ZHANG Shu², WANG Li-jun¹, BU Hong-jian¹

(1. Dept. of Software Engineering, University of Science and Technology of China, Hefei 230052;

2. Suzhou Institute for Advanced Study, University of Science and Technology of China, Suzhou 215123)

【Abstract】 P2P technique is an effective way to have resources shared in distributed and shared system. The efficiency of resource allocation is greatly enhanced based on P2P, but the download workload is not always balanced for some factors like the network bandwidth, the location of resource etc. As to solve this problem under the condition of multi-resource single user and multi-user single resource, a new method based on network bandwidth is proposed, through the resource provider self-discovery and self-request other available resource to achieve balanced distribution of resources load. It is proved by experiment.

【Key words】 P2P download; resource sharing; self-load-balancing

1 概述

P2P网络在信息共享领域发挥了重大作用。P2P不依赖或尽可能不依赖中央服务器,在一定程度上解决了C/S结构带来的单点失效、性能瓶颈和网络拥塞等问题,成为了互联网最为有效的共享与协作技术之一。在P2P信息共享系统中,如何在有限的带宽条件下将P2P资源分发性能提高到最大,这成为国内外专家们研究的重要目标之一,同时也取得了不少研究成果,例如分片算法、路由算法、流量算法等。这些成果在工程实践中给P2P技术带来了不同程度的发展。但是,随着网络环境的变化,P2P系统中各用户的带宽和负载能力也有较大的差异^[1-2],因此,如何平衡它们之间由负载能力所带来的差异,较好地实现无限资源通过有限的、负载能力不均的网络在用户间进行平衡分配,这正是本文要解决的问题。

本文从被下载者出发,通过被下载者自我感知自身负载,主动向其他 Peer 发出资源协作请求的方法来分配下载负载、疏通下载通道、减少下载堵塞、提升网络利用率和资源利用率和实现下载负载基本均衡。其基本原理描述如下:

A: 请求者, B: 资源拥有者, S: 索引服务器

Begin:

A 需要资源 K

A 通过 S 知道 B 具有资源 K

A 请求 B: 我需要 K 资源

B 回答 A: 同意

B 开始计算和分割资源

B 发现资源太大,而且 A-B 的网络带宽有限

B 向 S 询问还有没有其他人有这类资源?

S 查询到还有 X, Y 有 K 资源,然后告诉 B

B 则和 X, Y 协商将资源如何分割,三人合力将资源传给 A

X, Y 和 A 通信,看是否连通

假如 X 发现不能连接到 A,则 X 通告谁还能连接 A,代我传资源给 A

Z, F 告诉 X 我能连接到 A

X 计算 Z, F 到达 A 的带宽,看通过谁最快达到 A

X 选择到达 A 的路由,计算重新分配资源块

X, Y, B 根据和 A 的带宽分割资源

X, Y, B 向 A 传送资源块

END

基于上述原理,本文提出通过文件分割,资源块选择的方法来实现P2P系统中平稳、可靠的资源下载^[3]。

2 文件分割

在P2P系统中,资源分割指将同一资源按任务分割成等份或者不等份的文件块以分配给不同的资源提供者供用户下载。在目前的P2P系统中,资源一般是等份的,这样控制简单、方便,但是在不同带宽的用户中分配相同大小的下载块,必然引起部分用户饥饿,部分用户过载。对此,本文提出基于资源提供者和资源请求者之间的带宽来分割下载的资源的方法实现负载均衡^[4]。

为了避免分块过大或者过小而引起请求频率过高,分割时需要确定一个分割基数 B,该基数可以通过网络带宽选定,也可以根据文件大小来划分,同时也允许请求者自定义,其特性和规则如下:

(1)应为 16 的倍数,这样有利于转换为 Base64 字符串;

作者简介: 陶明亮(1979—),男,硕士研究生,主研方向:软件系统设计,计算机网络;张 曙,博士;王李军、卜红建,硕士研究生
收稿日期: 2006-11-24 **E-mail:** tml@mail.ustc.edu.cn

(2)大小不超过网络带宽最小的节点1分钟的下载量。

文件分割以分割基数 B 为基准,根据资源请求者和资源提供者间的网络带宽来划分文件块,分块原则为在以基准块基础上,增加分配负载量,直到负载量到达或者接近 Peer 的最大承受力,通过这样试探分配,逐步使得网络下载负载均匀,平稳地分布在不同的 Peer 之上。假设有 10MB 文件,资源提供者如表 1 所示。

表 1 资源提供者带宽

资源拥有者	到资源请求者带宽/(Kb·s ⁻¹)
A	20
B	32
C	100
D	50
E	60

分割基数 B 为 $1*60*20*1000/8=150KB$

资源的分块过程如下:

第 1 次分块,从文件头开始分,每个资源拥有者提供 1 分钟的数据量。还剩余 10240-1915=8325KB。如表 2 所示。

表 2 第 1 次分配

资源拥有者	块的大小/KB
A	2*150
B	2*240
C	2*750
D	2*325
E	2*450

第 2 次分块,增加 1 分钟的数据量,且从文件后面开始分。还剩余 8325-3830=4495KB。如表 3 所示。

表 3 第 2 次分配

资源拥有者	块的大小/KB
A	2*150
B	2*240
C	2*750
D	2*325
E	2*450

第 3 次分块,应该是 4 分钟的数据量,由于剩余的数据远少于 4 分钟的数据量,因此开始递减,分钟数为 $4/2=2$,选择文件块大小为 2 分钟数据量,从文件前面开始划分。还剩余 4495-3830=665KB。如表 4 所示。

表 4 第 3 次分配

资源拥有者	块的大小/KB
A	150
B	240
C	750
D	325
E	450

第 4 次划分,为第 3 次的一半即 1 分钟数据量,选择带宽最大的资源拥有者,看是否能一次提供剩余的数据:750KB > 665KB。如表 5 所示。

表 5 第 4 次分配

资源拥有者	块的大小/KB
A	0
B	0
C	665
D	0
E	0

分割算法如下:

```

Procure AllocateResource
Input Resource owners list,Resource Name
Output AllocationTable
Begin
//通过初始表计算 1 分钟数据量的块大小
OneMinuAllocation = CalculateOneMinuBlock(OriginalList)
PreviueAllocation = 0
Length = ResourceSize;

```

```

RestResource= Length;
Boolean fromHead = TRUR;
Int count = 1;
While Length >0
RestResource= RestResource- PreviueAllocation
Int NextAllocation = OneMinuAllocation * Count
IF RestResource < NextAllocation THEN
RestResource=RestResource+PreviueAllocation//复原
Count = count/2
Continue
END IF
IF Count >0 THEN
PreviueAllocation = GenerateAllocationResource(fromHead,
Count)//产生一次分配
fromHead = NOT fromHead
Length = Length -PreviueAllocation
Count = count * 2
ELSE
Sort Resource Owners List//增序排列资源带宽列表
FOR Int I=List.Count;I>=0ANDLength>0;I-- THEN
AllocateOneMinuResource(List[i]); //依次
//按带宽分配剩余的资源
Length = Length - List[i].Count
END FOR
END IF
End While
End

```

通过上述算法在多个资源副本、单个或者较少的下载者的情况下分配资源是有利的,它能快速、有效地实现 Peer 之间的负载均衡分配,实现动态平衡^[5]。

3 资源块选择

在单个资源、多个下载者情况下,上述文件分割策略需要改变,一般采用均分法分割资源,但是它不能保证下载者能均衡地得到他们所需要的资源,对此,本文采用对称分割的方法来分配资源块,算法基于的理论如下:

(1)理论上,只要 Peer 网络中有足够多的 Peer,且带宽足够大的话,则每一个 Peer 都可以成为其他下载者的 Agent,来转发由资源提供者所发出的数据包。

(2)网络带宽是相对的,因此,资源块下载的次序可以错开的,A 取得的资源块可能在某个时刻 B 需要它。

(3)资源块下载次序可以变化的,即拓扑排序不一定完全适合 Peer 网络中资源块下载排序^[6]。

(4)邻居(网络带宽达到一定值的 Peer)富,则自己富。

(5)最近邻居拥有了资源块,自身也能很快拥有,因此,可争取一切力量从邻居处取得自身没有的资源块。

(6)采用对称方法分块,即头尾拉锯式分配,这样能有效减少资源块的冲突,又能更好地实现 Peer 之间的资源互补^[7-8]。

为了理解方便,本文假定权值为一个时间单位内发送的资源块数。假设资源块为 10 块,A 为种子节点,最大连接数不限定,Peer 的资源请求顺序为: C, B, D。Peer 节点在一个单位时间内取得的资源块数是它到资源节点所有路径中的最大值。为了保证每个 Peer 都能最快地下载到自己所需要的资源块,Peer 节点请求的资源块需要错开请求。Peer 请求资源的目标 Peer 也是不固定的,它可以直接向资源提供节点请求,也可以向其他 Peer 请求,例如 B 节点可以向 C, D 或者直接向 A 请求资源块,同时它也能提供资源给其他 Peer,但是请

求块尽量避免有缺块或者冲突,因此,需要对资源块进行选择传送。本文选择跨步方法选择要传送的资源块,尽量减少资源块的连续性和内容相关性即资源块和内容无关,只和位置有关,因为离散的资源块可以大大减少相邻节点间的重复,增加在相邻节点取得资源块的机率。由于每个 Peer 既是资源请求者又是资源提供者,理论上这两者发生的概率是一样的,因此本文将跨步数选择为该 Peer 和其他 Peer 节点之间的权值的一半。跨步计算公式:

$$Step = \begin{cases} 1 & degree = 1 \\ \lfloor (\sum_{i=1}^{degree} Value-selfValue)/2 \rfloor & degree > 1 \end{cases}$$

selfValue:当前 Peer 到向它请求资源 Peer 的权值。

(1)首先 C 向 A 请求资源块, A 发送若干个资源块给 C, $Step_{AC}=1$ 。

(2)B 节点加入,它同时向 A, C 请求资源: $Step_{ab}=1$, $Step_{cb}=1$ 。C 也将向 B 请求资源: $Step_{bc}=1$ 。

(3)D 节点加入: $Step_{ad}=3$, $Step_{cd}=3$, $Step_{bd}=2$ 。

(4)C 将向 D 请求资源: $Step_{ac}=1$, $Step_{bc}=1$, $Step_{dc}=1$ 。

(5) B 将向 D 请求资源: $Step_{ab}=3$, $Step_{cb}=3$, $Step_{db}=2$ 。

计算 Step 时,基准块为当前 Peer 所缺少资源的最小块号,由于块的编号和资源的连续性没有关系,因此通过计算所请求资源的缺块率一般都很低;当发生缺块或者冲突时候, Step 值加倍即 $Step=2*Step$,如果 Step 值超过总块数,则可以通过取模的方法再次选择,但是当缺块个数非常少时,可以考虑依次请求缺块的方法(Step 为 1),这样尽量选择冲突概率小的资源块。通过该方法就可以使得每个 Peer 都能以最大能力取得和提供共享资源,实现下载负荷平衡分配。

4 实验分析

本实验资源如表 6 所示。

表 6 实验资源

单资源、多请求者实验环境配置		多资源、单用户实验环境配置	
网络环境	512Kb/s ADSL 宽带网	网络环境	512Kb/SADSL 宽带网
索引服务器	1 台	索引服务器	1 台
注册服务器	1 台	注册服务器	1 台
客户端机器	10 台	客户端机器	1 台
实验资源	10GB 数据文件	实验资源	10GB 数据文件: 2 台:相对客户机带宽 512Kb/s 1 台:相对客户机带宽 256Kb/s 2 台:相对客户机带宽 56Kb/s

由实验结果可知,在单资源、多请求者的情况下(图 1 的横坐标表示时间,从右到左增加),当资源分块数能满足请求者请求数时,下载速度线性增加,负载平稳分布在不同的 Peer 之上,当请求超过资源分配块的时候,以及在后期有缺块情况发生时,负载会有一段重新分配的过程,下载速度减慢,甚至出现循环等待的情况。但当某一用户下载完毕后,一个 Peer 的负载被卸载时,这种情况将被解除,经过重新分配资源块后,下载速度将上升。在多资源、单用户情况下(图 2,其横坐标表示时间,从右到左增加),下载负载随着资源的增加而急剧增加,但是整个分配过程平稳、均衡。当达

到客户端的最大带宽时,负载达到分配极限,不再进行分配,速度趋向平稳;若中途请求的资源中断,由于重新定位资源,寻找可以分配负载的 Peer,将在一小段时间内速度下降,这段时间由资源定位服务器决定。



图 1 单资源、多请求者客户端流量



图 2 多资源、单用户客户端流量

实验结果证明,不同的带宽对整个下载过程产生的影响不同,网络流量负荷基本上是按带宽均匀分布的。一个 Peer 断开,它的负载能快速、均衡地分布到其他 Peer 之上,使整个下载过程基本上是平稳和均衡的。

5 结束语

实验证明,在动态变化的网络环境中,采用文件分割策略和对称资源块选择方法能有效地在不同的 Peer 之间分配下载负荷,实现负载均衡下载。该方法有效地将 P2P 环境中一对多的关系转换为多对多的关系。对于初始状态就是多对多的情况,可根据现实情况选择任一方法,或者结合两者使用。

参考文献

- 王 珏. BitTorrent 下载技术研究[J]. 科技广场, 2005, (2): 26-28.
- 陈 姝, 方滨兴, 周勇林. P2P 技术研究和应用[J]. 计算机工程与应用, 2002, 38(13): 20-23.
- 陈 晗, 李志署. 基于 Web Service 的区域式 P2P 系统模型研究[J]. 计算机应用, 2004, 24(5): 18-21.
- 姚佳丽, 王 珊. 基于 P2P 数据索引和查询[J]. 计算机科学, 2005, 32(3): 69-72.
- 杨 宾, 孟 波. P2P 经典路由算法改进[J]. 计算机工程与设计, 2004, 25(2): 262-264.
- 郑 莹, 何初勇. P2P 文件共享系统中对等点发现机制的实现[J]. 信息技术, 2005, (10): 53-56.
- 彭 力. 一种新型对等网络文件共享系统设计[D]. 武汉: 华中科技大学, 2004.
- 何虎鸣. 基于 P2P 计算模型的局域网分布式存储系统[D]. 成都: 电子科技大学, 2003.