ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.18, No.1, January 2007, pp.105−116
DOI: 10.1360/jos180105

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

基于轨迹标签的无结构 P2P 副本一致性维护算法*

谢 鲲[1], 张大方[2+], 谢高岗[3], 文吉刚[1]

[1](湖南大学 计算机与通信学院,湖南 长沙 410082)

[2](湖南大学 软件学院,湖南 长沙 410082)

[3](中国科学院 计算技术研究所 信息网络研究室,北京 100080)

A Trace Label Based Consistency Maintenance Algorithm in Unstructured P2P Systems

XIE Kun[1], ZHANG Da-Fang[2+], XIE Gao-Gang[3], WEN Ji-Gang[1]

[1](College of Computer and Communication, Hu'nan University, Changsha 410082, China)

[2](School of Software, Hu'nan University, Changsha 410082, China)

[3](Network Research Division, Institute of Computing Technology, The Chinese Academy of Sciences, Beijing 100080, China)

+ Corresponding author: Phn: +86-731-8821980, Fax: +86-731-8821977, E-mail: dfzhang@hnu.cn, http://www.hnu.cn/rjxy

Xie K, Zhang DF, Xie GG, Wen JG. A trace label based consistency maintenance algorithm in unstructured P2P systems. *Journal of Software*, 2007,18(1):105−116. http://www.jos.org.cn/1000-9825/18/105.htm

Abstract: Replication is an effective way to improve the scalability, fault-tolerance, and availability as well as to reduce the query responding time in P2P system. With the P2P applications transferring from read-only static files sharing to read-write dynamical files interacting, maintaining consistency between frequently-updated files and their replicas is a fundamental reliability requirement for P2P system. This paper presents a trace label based consistency maintenance algorithm. It modifies the message datagram by attaching the address list of peers to which message has been sent. This can help to tell the duplicated message from the source peer by the aid of the attached address list in message datagram. Considering that the address list can become longer with the update time lapsing and the degree of P2P increasing, this paper presents a new Bloom Filter denoting the address list algorithm. The Bloom Filter can succinctly present the address list and simplify the query actions in the list by "OR" operations. The experimental results show that the new trace label based consistency maintenance algorithm can largely reduce the number of the duplicated messages. Moreover, the higher the degree of P2P, the more reduction of the number of duplicated messages and bandwidth utilization. The idea of consistency maintenance in this paper can also be applied to sensor network and other ad hoc networks.

Key words: consistency maintenance; unstructured P2P; trace label; Bloom filter; Gnutella

摘 要: 副本的存在是一种提高 P2P 系统的可扩展性、容错性、可用性和减少查询响应时间的有效手段.随着 P2P 应用逐渐由只读静态文件共享转换为需要实时更新的读写动态文件交互,副本一致性维护成为确保新业务

* Supported by the National Natural Science Foundation of China under Grant Nos.60473031, 60273070, 60403031, 90604015 (国家自然科学基金); the National High-Tech Research and Development Plan of China under Grant No.2005AA121560 (国家高技术研究发展计划(863)); the Natural Science Foundation of Hu'nan Province of China under Grant No.06JJ2090 (湖南省自然科学基金)

Received 2005-10-18; Accepted 2006-02-23

正确运行的关键.从直接更改消息报文角度出发,提出一种基于节点轨迹标签的无结构 P2P 副本一致性维护算法,通过在传输消息的报文中添加已接收更新消息的节点轨迹地址链表标签,可以在消息传输源节点进行冗余判断,以减少冗余消息数目.同时,针对直接存储节点地址轨迹标签算法的消息长度随着消息传输轮数和网络度数增加而不断加大的问题,提出一种用 Bloom filter 替代地址链表轨迹标签的算法.通过 Bloom filter 这种简洁的结构表示地址链表,可以减少添加到报文中的轨迹长度,利用 Bloom filter 的"或"运算可以简化传输节点的冗余判断.实验结果表明:节点轨迹标签算法可以极大地降低冗余消息数目,提高 P2P 系统的可扩展性.副本节点网络连通性越强,消息数目和传输带宽的减少就越明显.该研究可以用到传感器网络等其他自组织网络的一致性维护中.

关键词: 一致性维护;无结构 P2P 网络;节点轨迹标签;布鲁姆过滤器;Gnutella
中图法分类号: TP393 文献标识码: A

副本的存在是提高 P2P 系统的可扩展性、容错性、可用性和减少查询响应时间的有效手段.近年来,P2P 副本的研究重点集中在文件副本的建立策略和定位查找[1−4]上,以便合理地配置副本资源的位置,通过优化资源查找来保证网络的负载均衡.

然而,以往的研究认为,P2P 系统中共享的是静态资源,没有经常需要更新的内容,对这些资源的访问通常是只读形式,如 MP3,Video 等.随着 P2P 的新型应用,如 P2P 游戏、信任管理、目录服务、联机拍卖、远程协作等的出现,共享的文件以只读的形式转换为可读写形式,文件更新频繁,确保其副本的一致性是保证其正确运行的关键.因此,设计算法来维护 P2P 网络多个副本资源的一致性,保障 P2P 新业务的开展,是十分迫切的事情.一方面,如果没有有效的一致性维护算法,P2P 应用就只能局限于提供静态的非频繁更新的文件共享;另一方面,新业务的开展需要一致性的算法来为动态内容的更新提供保障.

目前,针对 P2P 副本一致性的研究并不是很多.在结构化 P2P 网络中(如 Chord,CAN 等),各节点的连接具有某种规则的结构,通过特殊设计的算法来严格控制数据存放和网络拓扑,一致性维护算法往往可以借助分布式 Hash 表完成;在无结构化的 P2P 网络中(如 Gnutella,Freenet 等),由于这类系统由大范围的自愿节点组成,没有对拓扑结构和文件存放的控制,对于数据的可用性和持久性通常只提供松散的保证.基于洪泛的副本一致性维护算法[5]、基于谣言(Rumor 或 Gossip)[6]的副本一致性维护算法等实现简单,但洪泛法会在网络中产生大量的冗余信息,而 Gossip 在减少冗余信息和所有副本都得到一致性维护之间需要权衡.此外,副本链[7]一致性维护算法可以有效地减少冗余信息,但需要额外构造副本链,而副本链的建立和维护都比较困难.

本文研究 P2P 网络中的副本一致性更新算法,针对现有的无结构分散 P2P 系统的副本更新算法冗余消息多、影响系统扩展性的问题,提出了一种基于节点轨迹标签的更新算法.在更新消息传送过程中,将每轮传输的目标节点集添加到消息报文头部作记录,通过记录已经获得更新的节点来阻止消息报文在已传播的节点中再次传播,从而减少更新的代价.基于这种思想,我们改进了洪泛法和谣言(gossip)算法;同时,针对直接存储节点地址轨迹标签算法的消息长度随着消息传输轮数和网络度数增加而不断加大的问题,本文提出用 Bloom filter 替代地址链表的轨迹标签.利用 Bloom filter 这种简洁的数据结构表示地址链表,有效减少了附加到消息报文的地址信息长度,同时简化了传输节点的冗余判断.理论分析和实验结果表明:改进的算法可以使副本消息的传播数量大为减少,从而有效地提高了 P2P 系统的可扩展性.

1 相关工作

副本的存在是一种有效提高 P2P 系统可扩展性和可用性的手段.就目前的研究来看,P2P 系统副本的一致性维护并没有得到广泛研究,这给建立持续可扩展的 P2P 系统带来了挑战.目前存在的无结构分散式 P2P 副本一致性维护的算法有:

Gnutella[5]采用基于洪泛的副本一致性维护算法.更新初始化节点将消息通过广播的方式告诉邻节点,其邻居节点再将此消息转发到它的下一轮邻居节点,以这种类似广度优先搜索的方式将更新消息传送到网络中所

有的副本.算法实现简单,可达到所有副本的完全一致性,但其传递消息的报文个数被指数级放大,占用网络大量带宽,一个副本节点多次收到更新消息,冗余较大.因此,这类系统大多采用为消息报文增加“跳步”限制的方法,即为每个消息报文设定一个 Time-to-Live(TTL)值,随着跳步的增加,*TTL* 减少,当 *TTL* 为 0 时,更新消息停止传播.由于 *TTL* 的选择具有盲目性,*TTL* 过大时 *TTL* 失去意义,产生大量的广播信息;如果 *TTL* 过小,广播覆盖范围有限,副本节点得不到有效的更新,降低了 P2P 的可用性.这也是这类系统缺乏扩展性的原因之一.

文献[6]提出谣言(gossip)“推(push)”和“拉(pull)”混合一致性维护算法.“推”方法类似于社会学中的谣言传送过程.基于谣言的“推”算法在每轮传输时,节点总是选择其部分邻居节点作进一步转发.与洪泛法相同,基于谣言的“推”都是由更新初始化节点主动发起更新消息的传递,主动将更新消息,“推”向其他副本节点.但是,当 P2P 网络中节点动态地加入和离开时,仅仅使用“推”的机制传递更新消息不再适应动态环境,因此,“拉”作为“推”的补充也用到一致性维护中来:当新的节点加入 P2P 网络时,新的节点通过主动连接邻节点来请求获得最新的副本,亦即 Pull 过程.文献[6]针对 P2P 网络动态环境提出了“推”、“拉”结合的一致性维护算法.基于谣言传递的“推”算法虽然可以减少部分冗余开销,但是其达到的一致性并非完全一致性,Gossip 在减少冗余信息和所有副本都得到一致性维护之间需要权衡.

文献[8]提出了一种主节点“推”与动态时间“拉”相结合的副本一致性维护算法.只有主节点(master)才能够作为更新消息的初始发起者,这对于 P2P 节点的更新是一种限制,当主节点的 IP 改变或者离线时,一致性维护就会失效.

文献[7]提出了一种基于副本链(replica chain)的副本一致性维护算法:更新消息通过副本链传递,网络中每个节点是副本链中的一个点,每次更新消息传递给链中邻近的 k 个节点.这种方法可以有效地减少冗余消息的产生,但是每个节点必须维护一个周围节点的副本链,而在无结构分散 P2P 网络中,节点只知道其直接相连的邻节点,构造和维护一个副本节点链带来了额外消息交互开销,而且算法性能参数 k 的选择也如洪泛法中 *TTL* 选择一样是一个折衷的结果.

文献[9]借助结构化 P2P 网络的分布式 Hash 表(DHT)建立一个“副本分割树”,获得每个副本的位置来传输更新消息.虽然该算法能有效地进行副本维护,但是,借助 DHT 的算法只适合结构化 P2P 网络.

除此之外,一致性维护中传播消息也有两种方式:一种方式只传送更新提示,由副本节点自己决定是否下载更新;另一种方式是直接传送更新给副本节点.

上述的研究表明,现有算法要么传输消息的冗余量大,要么需要另外构造表现网络的拓扑信息结构来辅助更新消息传输,参数确定困难,部分算法的实现也有限制;而且判断更新是否冗余是在消息报文传播之后,不能在传输的源头进行控制,不便于 P2P 网络的扩展.本文从传播报文的角度出发,在报文头部增加一个已经收到更新的节点轨迹标签,对已经传输的节点进行记录,收到消息的节点通过检查报文中的节点轨迹标签,完成对冗余消息的提前检测,减少消息在节点间的冗余传输,同时给出算法传输过程的理论分析也是本文的一个特色.

2 问题描述和相关定义

无结构分散式 P2P 副本一致性维护问题是指:在无结构分散式 P2P 网络中,一个或者多个副本节点由于动态读写原因,当副本出现改动时,通过节点间消息传递,确保相关节点保存的文件副本一致性.

在讨论一致性维护算法之前,以 Gnutella 为代表的无结构分散式 P2P 网络为例,给出以下假定:

假定 **1**. 各副本节点的更新传输稳定同步进行.

假定 **2**. 更新消息的大小相同.

假定 **3**. 每个副本节点都有一个对应的 ID 和 IP 地址.

假定 **4**. 节点可以随机地加入和离开系统.

假定 **5**. 节点只要知道其他节点的 IP 地址,便可以进行通信连接.

假定 **6**. 节点只知道与其直接相连的邻居节点的信息.

假定 **7**. 节点可以通过传递消息确定是否为某资源的副本节点.

定义 **1**. 副本节点 v 是指对某一个文件,拥有该文件的 P2P 节点(replica peer).

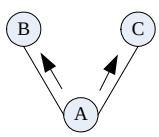
定义 **2**. P2P 网络中节点 v 的度 d(或邻居数)是指此节点相连的邻居节点个数.

定义 **3**. 平均每个副本节点在一次一致性维护中转发消息的个数为副本传输开销[5,10]:

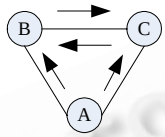
$$f=\left(\sum_{i=1}^{N}m_i\right)\Big/R \tag{1}$$

其中,R 为更新消息可达的节点个数;m_i 为节点 i 转发的更新消息个数;N 为拥有副本的系统规模(节点个数).

由式(1),在图 1(a)中,初始更新节点为 A,进行一次一致性维护副本传输开销为 0.6,图 1(b)所示的开销为 1.3.



(a) Replica update cost is 0.6
(a) 副本传输开销为0.6



(b) Replica update cost is 1.3
(b) 副本传输开销为1.3

Fig.1 Replica update cost

图1 传输开销示例

定义 **4**. 副本相对传输开销是指某算法的副本传输开销与洪泛算法的传输开销之比.

定义 **5**. 设拥有副本的系统规模为 N,在某一次一致性维护下副本更新消息可达的节点数为 R,则在该算法下的系统覆盖度 C 定义为

$$C=R/N \tag{2}$$

在 Gnutella 系统中,由于副本的传递和更新都是通过基于洪泛(flooding)机制,当节点收到更新消息时,如果该消息是第一次到达,则将消息转发给其所有的除消息来源以外的邻居节点,系统中节点的平均邻居数为 k,由式(1),其广播开销 $f_{flooding}$ 为

$$f_{flooding}=\left(1+\sum_{i=1}^{N}(k_i-1)\right)\Big/N=1/N+\sum_{i=1}^{N}(k_i-1)/N\approx k-1 \tag{3}$$

其中,k_i 为节点 i 的度数,即邻居数;N 为系统规模.同时,如果不考虑 TTL 的因素,其覆盖度 $C_{flooding}$=1.

定义 **6**. 进行一致性维护的时间是指更新从初始节点开始,到算法完成,传输的最大轮数,T 定义为

$$T=\max(t_j) \tag{4}$$

t_j 表示一条传输链的传输轮数.如图 1(a)所示,存在两条传输链 A→B 和 A→C,t_1=1,t_2=1,那么此时,T=1;如图 1(b)所示,存在两条传输链 A→B→C,A→C→B,t_1=2,t_2=2,此时,T=2.

定义 **7**. 当副本节点收到更新消息后,再次收到同版本的更新消息为冗余更新消息.

定义 **8**. 平均每个副本节点在一次一致性维护中收到的冗余副本更新消息的个数为副本冗余传输开销:

$$D=\left(\sum_{i=1}^{N}m_i-(R-1)\right)\Big/R \tag{5}$$

其中,R 为更新消息可达的节点个数;m_i 为节点 i 转发的更新消息个数;N 为拥有副本的系统规模.

在图 1(a)中,平均冗余传输开销为 0,系统收到的冗余消息数为 0;在图 1(b)中,平均冗余传输开销为 0.6,其收到的冗余消息数为 2.对于 Gnutella 的 P2P 网络来说,由式(5),其副本冗余传输开销为

$$D_{flooding}=\left(1+\sum_{i=1}^{N}(k_i-1)-(N-1)\right)\Big/N=2/N+\sum_{i=1}^{N}(k_i-1)/N-1\approx k-2 \tag{6}$$

评价一致性维护算法的指标有:副本传输开销、副本冗余传输开销、覆盖度和一致性维护时间.现有一致性维护算法的副本传输开销较大,占用了大量的带宽,所以如何减少副本传输开销是本文考虑的重点.

3 算法的提出

Gnutella 中采用洪泛法进行一致性维护.图 2(a)是它的一个实例,节点①是更新的发起者,第 1 轮传输时,节

点①向其邻节点发送更新消息:①→⓪,①→②,①→③,①→④;第 2 轮传输时,节点⓪,②,③,④向其邻节点发送消息:⓪→②,⓪→④,②→⓪,②→③,②→④,②→⑤,③→②,③→④,③→⑤,④→⓪,④→②,④→③,④→⑤;第 3 轮传输时,节点⑤向其邻节点发送消息:⑤→③,⑤→④.3 轮传输过后,共产生更新消息 19 条,副本冗余传输开销为 2.3,平均每个节点收到两条重复的更新消息.

大量冗余消息的产生,来自盲目的洪泛,该算法中每轮的传输都只排除一个节点(消息来源节点).如在第 2 轮中,节点⓪,②,③,④都是节点①的邻居节点,是节点①的直接近邻.它们之间的互传消息:⓪→②,⓪→④,②→⓪, ②→③,②→④,③→②,③→④,④→⓪,④→②,④→③都是冗余消息.

如何减少这种近邻之间的消息冗余?我们的灵感基于这样一个事实:如果邻接双方知道了对方已经接收到更新的消息,就不再发送更新消息给对方.如果能够记录这些已经接收更新的节点,并将该信息放在报文头中发送给其他节点,使其他节点不再发送更新消息到已经记录过的节点,就可以大大减少消息冗余,这就是本文算法的出发点.根据假定 6,每轮传输时,可以在传输的消息报文中预留一个节点轨迹标签,将每次发送的目标邻居节点集记录在此标签中,收到消息后,首先检查节点轨迹标签,更新消息只发往不在标签中的邻节点,这样就可以避免消息在近邻节点间的冗余传输.如图 2(b)所示,节点①向其邻节点⓪,②,③,④发送副本更新消息,第 1 轮,节点①将发送目标节点集和自身{⓪,②,③,④,①}作为节点轨迹添加到更新报文头部,第 1 轮产生更新消息:①→⓪,①→②,①→③,①→④;第 2 轮传输,节点⓪,②,③,④收到更新消息后,首先检查其邻节点是否已记录在节点轨迹标签,只向不在标签中的节点发送更新消息:②→⑤,③→⑤,④→⑤,同时在标签中记录新的节点⑤.这样,第 2 轮中减少更新消息 10 条;节点⑤收到的更新消息标签中已包含了其所有邻居节点,所以不再传输,这样减少 2 条更新消息.虽然利用节点轨迹标签算法的思路简单,却可以有效减少传输冗余,而且可以减少一致性维护时间(如:最后一轮,节点⑤不再继续传输更新消息给其他节点).

(a) Flooding based consistency maintenance algorithm (b) Trace label based consistency maintenance algorithm

(a) 基于洪泛一致性维护算法 (b) 基于节点轨迹标签一致性维护算法

Fig.2 Flooding and trace label algorithms

图2 洪泛算法和节点轨迹标签算法

基于节点轨迹标签的一致性维护算法的消息报文如图 3 所示.第 1 部分是添加的轨迹标签节点地址链表,第 2 部分是副本更新消息.

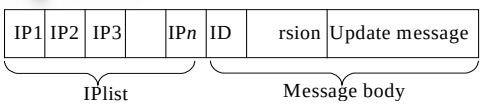


Fig.3 Message structure with trace label

图3 包含轨迹标签的消息报文

算法的思想是:在更新消息报文头部中添加已经传输的节点地址信息,节点在每发起新一轮的传输时,首先检查自己的邻居节点是否已经在更新消息的节点轨迹标签中,如果已经存在,说明此邻居节点已经得到更新消息,不向此节点发送,否则就向此节点发送,如图 4 所示.该算法具有以下优点:首先,节点轨迹标签保存了更新消息传输的轨迹,可以杜绝消息传输循环;其次,保证了近邻之间的消息不传递;最后,消息发送源节点可以主动判断来避免产生消息冗余,减少传输带宽.

```
IP_List_Algorithm (Message, peer)
//process the update message if the peer firstly receives it
If not (received (Message)) then
      Peer.IsReceived = True
      oldIPList = Message.IPList
      //add all neighbor addresses to the trace label
      For (every neighbor in peer.neighbor)
         Message.IPList += neighbor
      End for
     //send the update message to the neighbors not in the trace label
      For (every neighbor in peer.neighbor)
         If neighbor not in oldIPList
            push (Message, neighbor)
          End If
        End For
End If
```

Fig.4 Trace label based consistency maintenance algorithm

图 4 节点轨迹标签的副本更新算法

3.1 Bloom filter表示节点轨迹标签的一致性更新算法

虽然节点轨迹标签的副本更新算法可以减少消息传播数量,但是,由于在报文的头部增加了节点地址信息,消息报文变长,添加的地址链表长度是网络节点的度数和消息传播轮数的递增量.为了减少地址链表的长度,改进直接存储节点地址链表的轨迹标签算法,本文设计了一种用 Bloom filter 表示地址链表的节点轨迹标签更新算法.Bloom filter[11−13]是用来表示集合、支持集合元素查询的一种简洁结构,它对集合中元素的表示只需要少数几个比特.利用 Bloom filter 替代传输地址链表,消息报文如图 5 所示:一部分是用固定长度的 0,1 串表示的地址链表,另一部分是副本更新消息.

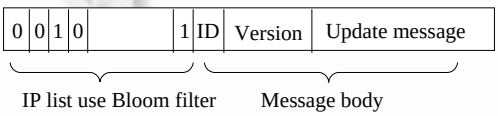


Fig.5 Message structure with Bloom filter denoting the trace label

图 5 用 Bloom filter 表示的轨迹标签的消息报文

消息报文用固定长度的 Bloom filter 结构(0,1 位串)代替原来的 IP 地址链轨迹标签,同时,每个节点保存一个用同样长度 Bloom filter 表示的节点地址掩码.判断一个邻居点是否已经接收过更新,只需将该邻居节点地址掩码和轨迹掩码进行“或”运算[12],如图 6 所示,得到新的轨迹掩码.如果新的轨迹掩码与原轨迹掩码不同,表明该邻居没有记录到标签中,就将更新消息发到该邻节点.

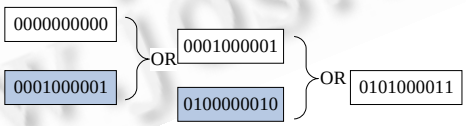


Fig.6 Bloom filter “OR” operation

图 6 Bloom filter“或”运算

用 Bloom filter 结构代替地址链表,有两点好处:1) 添加到消息报文的附加信息减少;2) 检查节点是否在轨迹标签过程运算十分简单,只需要“或”运算.但是,用 Bloom filter 代替直接传送的地址链会产生一定的误差,这是由于 Bloom filter 查询会出现少量的假阳性错误,导致判断时邻节点因没有接收更新消息而被误认为已经接收而减少了更新的覆盖度.下面分析 Bloom filter 查询所引起的假阳性概率对算法的影响.式(7)是 Bloom filter 假阳性概率[11,13].

$$f=(1-p)^k=\exp(k\cdot\ln(1-e^{-kn/m})) \quad (7)$$

k 为每个节点地址需要映射到 Bloom filter 中的位数;n 为节点数;m 为 Bloom filter 长度.可能产生的判断错误节点数为

$$E=f\cdot n=\exp(k\cdot\ln(1-e^{-kn/m}))\cdot n \tag{8}$$

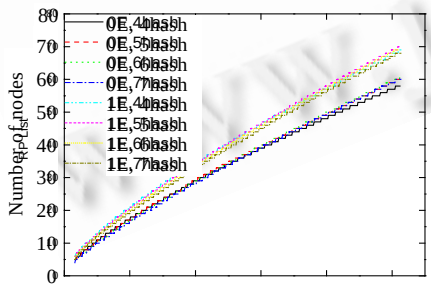
```
Bloom_IP_List_Algorithm  (Message, peer)
//process the update message if the peer firstly receives it
If not (received (Message)) then
   Peer.IsReceived = True
   oldTraceBloomFilter = Message.TraceBloomFilter
   //add all neighbor addresses to the trace label denoted by Bloom Filter
   For (neighbor in peer.neighbor)
      Message.TraceBloomFilter |= neighbor.BloomFilter
   End for
    //send the update message to the neighbors not in the trace label
   For (neighbor in peer.neighbor)
       temp_Bloom = oldTraceBloomFilter
      oldTraceBloomFilter |= p_neighbor.BloomFilter
      If (temp_Bloom == oldTraceBloomFilter)
          Then  MightHaveReceived (neighbor)
           Else   Push (Message, neighbor)
      End if
   End For
End IF
```

Fig.7 Consistency maintenance algorithm with trace label denoted by Bloom filter

图 7 用 Bloom filter 表示节点轨迹的一致性维护算法

用图 8 来评估查询误判断的影响,横坐标是代表 Bloom filter 的长度,纵坐标表示用该长度的 Bloom filter 表示的节点个数.曲线表示当出现 E 个判断错误前,用横坐标长度的 Bloom filter 可以表示的节点个数.曲线 0E, 4hash 表示当节点地址用 4 位映射不出现判断错误节点的曲线,如曲线上点(36,9)表示用 36 位 Bloom filter 可以表示 9 个地址不会出现误判断.通过对 Bloom filter 产生错误判断进行评估,使用时,只需要根据可能的节点长度选择相应长度的 Bloom filter,就可以保证不发生节点查询的误判断.

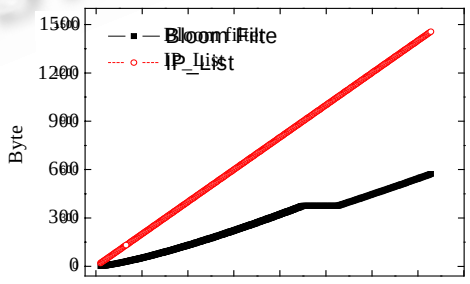
使用 Bloom filter 可以减少附加到报文的节点地址信息长度:假设节点规模 N=1000,平均节点度 k=10,传输 6 轮后,按最坏的情况考虑,此时需要标志的节点为 60 个节点,需要 512 位(64 字节)Bloom filter 表示.若直接采用 IP 地址存储同样多的节点地址链,需要 240 个字节,用 Bloom filter 减少了附加到报文中的地址信息长度,而且判断 IP 地址是否在地址轨迹标签的操作更为简单(“或”运算).图 9 比较了 Bloom Filter 和直接用地址链表表示轨迹标签算法所需存储结构长度(其中,Bloom filter 的长度取图 8 中不出现判断错误时的长度),横坐标表示需要附加到报文的节点地址个数,发现随着节点个数的增加,Bloom filter 表示的轨迹标签与直接用 IP 地址链表表示的轨迹算法相比,加到报文的标签轨迹长度明显减少.



Bloom filter bits

Fig.8 False positive evaluation of Bloom filter

图8 Bloom filter误判评估



Number of nodes

Fig.9 Storage comparison of Bloom filter and IP_List

图9 Bloom filter和地址轨迹标签所用空间比较

3.2 节点轨迹标签改进的Gossip算法

上述讨论是基于洪泛算法而展开,我们同样可以将节点轨迹标签思想应用到谣言法之中.谣言传输法由于

其选择一定概率的子集进行类似流言的副本消息传输,可以降低消息传输的数量而受到关注.然而,Gossip 算法并没有记录已收到消息的节点.用节点轨迹标签改进 Gossip,将每轮选择进一步传输的节点子集记录到传递的报文头中,这样也可以在 Gossip 传输中避免消息传递循环和近邻之间的冗余消息.其中,f 是 Gossip 算法中任一节点选择下一轮传输的邻居节点子集的比率.由于篇幅所限,具体算法描述从略.

4 算法理论分析

假设 P2P 网络中有 n 个副本节点,节点平均度数为 k,每条边以概率 p 独立存在:

$$p=k/(n-1) \tag{9}$$

在算法分析中,我们讨论洪泛法和节点轨迹标签更新算法的传播速度和消息数目.对于 Gossip 和改进 Gossip 算法,限于篇幅,这里不作详细讨论.

第 1 轮传输.更新消息从 R_0 出发,$FR(0)=1$,通过洪泛法向 k 个邻居节点发送更新消息:

$$FMsg(1)=k \tag{10}$$

第 1 轮传输后,新增收到消息节点为 $FNewR(1)=k$,此时,共有 $FR(1)=1+k$ 个节点收到了更新消息.

第 2 轮传输.根据洪泛法的消息传输机制,第 1 轮中新收到消息的节点向除 R_0 节点外的邻居节点发送广播消息:

$$FMsg(2)=k\cdot(k-1) \tag{11}$$

第 1 轮传输后,新增节点集为$(m_1,m_2,\ldots,m_k)$,第 2 轮传输由新增节点集中每个节点向除 R_0 之外的相连的 $k-1$ 个节点发送传输消息,m_1 发送消息后,新增加的节点个数为 $A_1=(k-1)-(FNewR(1)-1)\cdot p=(k-1)(1-p)$,第 m_k 节点发送消息增加的节点为

$$A_k=(k-1)-(FNewR(1)+A_1+\ldots+A_{k-1}-1)\cdot p=(k-1)(1-p)^k \tag{12}$$

第 2 轮传输中,新增加的节点为

$$FNewR(2)=A_1+A_2+\ldots+A_k=(k-1)(1-p)+\ldots+(k-1)(1-p)^k=(k-1)(1-p)(1-(1-p)^k)/p \tag{13}$$

第 2 轮传输后,收到更新消息的节点有

$$FR(2)=FR(1)+FNewR(2)=1+k+(k-1)((1-p)-(1-p)^{k+1})/p \tag{14}$$

第 t 轮传输($t\geq 3$).第 t 轮传输的消息数、新增获得更新消息的节点和更新消息节点总数分别为

$$FMsg(t)=FNewR(t-1)\cdot(k-1) \tag{15}$$

$$FNewR(t)=(k-1)(1-p)^{FR(t-2)}(1-(1-p)^{FNewR(t-1)})/p \tag{16}$$

$$FR(t)=FR(t-1)+FNewR(t) \tag{17}$$

t 轮传输后共发送消息个数为

$$FMsgTotal(t)=FMsg(1)+FMsg(2)+\ldots+FMsg(t)=(k-1)^2(1-p)(1-(1-p)^{FR(t)-1})/p \tag{18}$$

由洪泛法和节点轨迹标签算法的特点可以得出:$NewR(i)=FNewR(i)(1\leq i\leq t)$.类似地可以推导出,在节点轨迹标签算法中,经过 t 轮传输共发送消息个数为

$$MsgTotal(t)=Msg(1)+\ldots+Msg(t)=k+NewR(1)((k-1)-(Record(1)-1)\cdot p)+\ldots+$$
$$NewR(t-1)((k-1)-(Record(t-1)-1)\cdot p) \tag{19}$$

节点轨迹标签一致性算法比洪泛算法减少的消息数为

$$FMsgTotal(t)-MsgTotal(t)=NewR(1)(Record(1)-1)\cdot p+NewR(2)(Record(2)-1)\cdot p+\ldots+$$
$$NewR(t-1)(Record(t-1)-1)\cdot p \tag{20}$$

从式(20)发现:$NewR(i)(1\leq i\leq t)$,$Record(i)(1\leq i\leq t)$,p 都是 k 的递增函数.k 越大,图的连通性越强,基于节点轨迹标签的更新算法,节约传递的冗余消息数就越多.

表 1 为相关的算法理论分析说明.

Table 1 Description for theory analysis

表 **1** 算法理论分析说明

R_0	The initial update peer
$FMsg(t)$	Number of update messages in round T of Flooding algorithm
$FR(t)$	Number of updated replica peers after round T of Flooding algorithm
$FNewR(t)$	Number of updated replica peers in round T of Flooding algorithm
$FMsgTotal(t)$	Number of update messages after round T of Flooding algorithm
$R(t)$	Number of updated replica peers after round T of Trace label algorithm
$NewR(t)$	Number of updated replica peers in round T of Trace label algorithm
$Record(t)$	Length of the IP_List included in update message in round T of Trace label algorithm
$MsgTotal(t)$	Number of update messages after round T of Trace label algorithm

5 实验仿真

进行仿真时,必须产生具有对等网络特性的网络拓扑结构.P2P 网络的一个重要特征是节点度服从幂律分布,其节点度为 k 的节点的分布概率满足 $P(k)\propto k^{-\tau}$,其中,$1<\tau<\infty$,网络中少数节点有较高的度.通常,把节点度服从幂律分布的网络称为无标度网络(scale-free network),并称这种节点度的幂律分布为网络的无标度特性.1999 年,Barabási 和 Albert[14]给出了构造无标度网络的演化模型,他们把真实系统通过自组织生成无标度网络归功于两个主要因素:生长本质和偏好依附,BA 模型就是根据这两个因素构造的网络拓扑,服从幂律分布规律.实验中采用波士顿大学开发的 BRITE[15]来产生 BA 网络模型.同时考察到 P2P 网络中副本节点规模随着文件副本放置策略和文件的热点程度而有不同的文件副本节点覆盖率[2,3,16,17].就 Gnutella 网络来说,80%的请求指向的文件的副本数目在 80 以上[17],副本的平均覆盖率为网络总节点数的 1%[16],当前系统的副本规模大约为 100 节点,本文取副本节点规模为 100~1000.

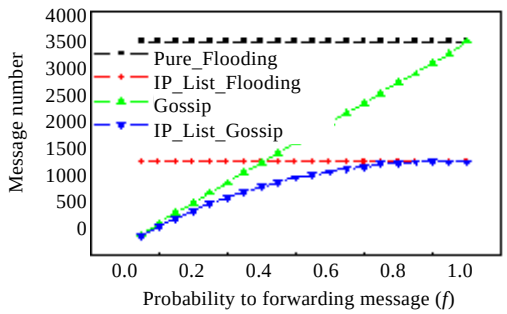
更新开始节点的选择,会产生实验结果的差异.所以在实验中,将网络所有节点作为更新的初始化节点进行仿真实验,最后的实验结果采用各节点作为初始化节点传输的平均值来屏蔽由于选择节点而带来的算法差异.

5.1 仿真实验

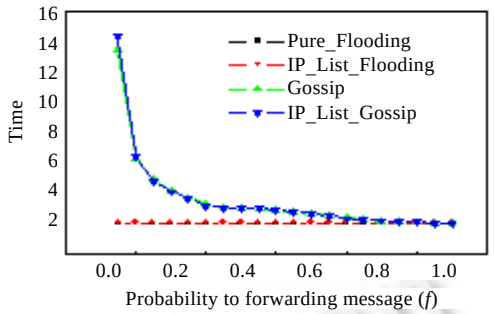
图 10 是用 BRITE 产生的节点规模 N=100,平均度为 d=20 的 BA 网络模型的 4 种算法比较图,横坐标是使用 Gossip 算法每轮选择进一步转发的节点比率 f.显然,洪泛算法和基于节点轨迹标签的一致性维护算法没有这个参数 f,但是为了比较方便,我们还是画到图中,用直线表示.可以发现:当 f 过小时,Gossip 算法的覆盖度比较低,所以 f 过小的 Gossip 算法在实际中是不可取的.基于节点轨迹标签的一致性维护算法可以获得和洪泛法一样的覆盖度,且传输的时间可能会较少(如在实验 N=10、平均度为 d=6 时,发现 $T_{flooding}$=3 而 T_{IP_List}=2,这是因为:虽然洪泛法和节点轨迹标签法的更新传输扩展速度一致,但是由于节点轨迹标签算法记录了前面的已经传输的节点,可以进一步减少更新时间),副本传输开销大为减少,这是节点轨迹算法最大的优点.

图 11 是用 BRITE 产生的节点规模 N=100,网络不同度数情况下的算法比较,横坐标是节点的平均度数,纵坐标是各种算法的副本传输开销.因为纯 Gossip 算法在 f=1 时可以等同于洪泛算法,而 f 太小其覆盖度有限,因此取 f=0.8 和 f=0.6.我们发现:基于节点轨迹标签的算法具有较低的副本传输开销,当平均度数为 90 时,使用节点轨迹标签算法更新消息数仅占洪泛算法的 1.9%,占 f=0.6 时 Gossip 算法消息数的 3.1%,极大地减少了副本更新的消息数量.用节点链表改进的 Gossip 算法也具有相似的低传输开销性能.

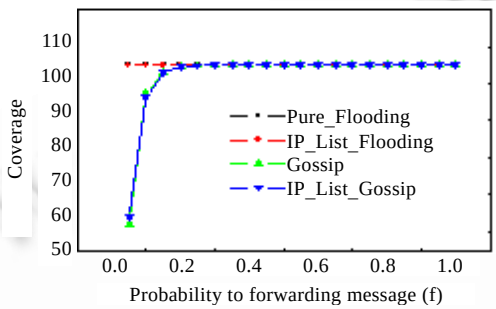
图 12 是用 BRITE 产生的节点规模 N 从 100~1000、节点度 d=20 的 BA 网络模型,考察的是副本传输开销与系统规模的关系,横坐标代表的是节点规模,纵坐标是各种算法的副本传输开销.使用基于节点轨迹标签改进的 Gossip 算法的副本传输开销最小,当 f=0.6,N=1000 时,其传输开销比洪泛法减少 49.3%,比纯 Gossip 算法减少 15%;当 f=0.6,N=100 时,基于节点轨迹标签改进的 Gossip 算法的副本传输开销比洪泛法减少 65.6%,比纯 Gossip 减少 41.7%,远大于节点规模 N=1000 时节约的百分比.这是因为 N=100,d=20 时图的连通性比 N=1000,d=20 时的连通性要强,所以节点轨迹标签算法在副本传输开销上的改进也就越明显.



(a) Replica update cost
(a) 副本传输开销



(b) Consistency maintenance
(b) 一致性维护时间



(c) Coverage
(c) 覆盖度

Fig.10 Algorithm comparisons under N=100, d=20 BA topology
图 10 N=100,d=20 时 BA 拓扑结构中 4 种算法比较

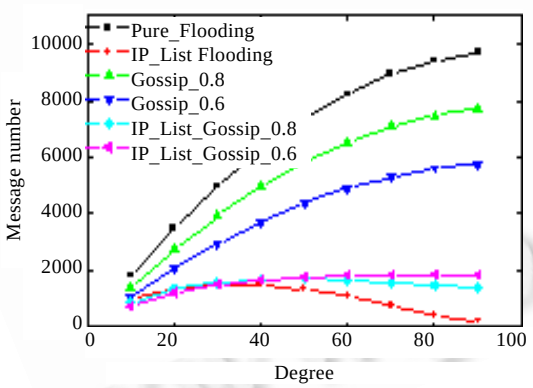


Fig.11 Replica update cost with system degree
图 11 副本传输开销与度数的关系

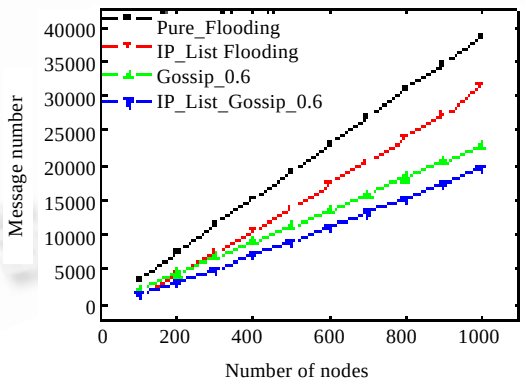
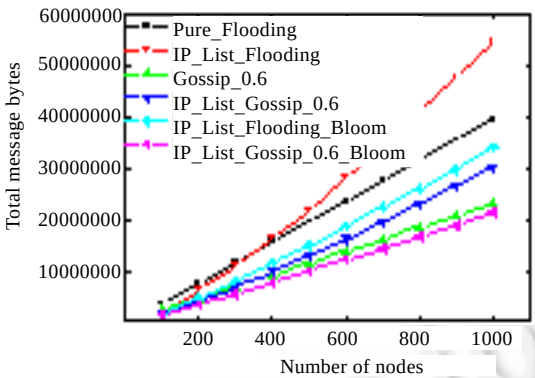


Fig.12 Replica update cost with the number of nodes
图 12 副本传输开销与系统规模的关系

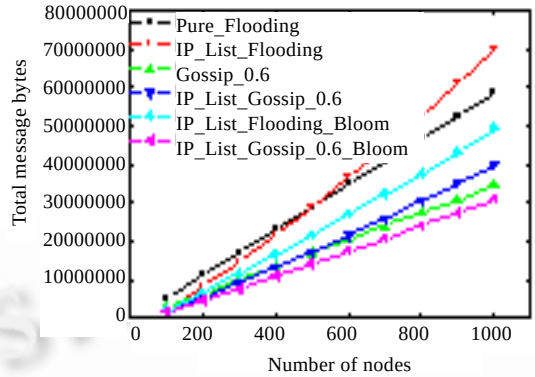
图 13 是用 BRITE 产生的节点规模 N 从 100~1000、节点度 d=20 的 BA 网络模型,比较更新消息传输的总字节数.横坐标代表的是节点规模,纵坐标是各种算法的传输总字节数.

图 13(a)~图 13(c)分别比较了消息数据包长度为 1 024,1 518 和 5 000 字节时,4 种算法的更新消息的总字节数.直接使用地址轨迹标签算法,由于捎带的地址节点数目较大,传输的总字节数较大,而用 Bloom filter 表示节点轨迹标签改进的 Gossip 算法的消息传输总字节数最少,可大大节约传输更新消息的带宽.当更新消息数据长度为 5 000 字节、节点规模 N=1000,f=0.6 时,使用 Bloom filter 表示节点轨迹标签改进的 Gossip 算法附加到数据报文的节点地址链表总字节比直接用 IP 地址表示的轨迹标签算法减少 91.9%;所传输消息总字节数比洪泛法减少 51.3%,比纯 Gossip 算法减少 13%.Bloom filter 表示节点轨迹标签改进的 Gossip 算法在副本传输开销(如

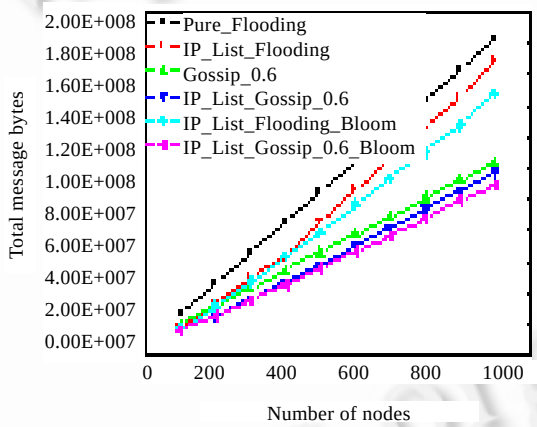
图 12 所示)和传输消息总字节数比纯 Gossip 算法减少的百分比相当,这 2%的微弱差距是因为添加 Bloom filter 标签引起的代价开销.当 N=100,f=0.6 时,Bloom filter 表示节点轨迹标签改进的 Gossip 算法传输总字节比 Gossip 减少 40.9%,远大于 N=1000 时节约带宽的百分比,这是因为 N=100,d=20 时图的连通性比 N=1000,d=20 时的连通性要强,所以,节点轨迹标签算法在消息传输总字节数(即占用网络带宽)上的改进也就越明显.



(a) Message body is 1 024 bytes
(a) 消息长度为 1 024 字节



(b) Message body is 1 518 bytes
(b) 消息长度为 1 518 字节



(c) Message body is 5 000 bytes
(c) 消息长度为 5 000 字节

Fig.13 Comparisons of the total message bytes
图 13 消息传输总字节比较

6 总 结

本文主要讨论无结构分散式 P2P 网络一致性更新算法,发现洪泛法和谣言算法由于对已经更新过的节点没有记忆性,因而产生了大量的消息冗余.本文研究从新的视角来控制一致性维护算法的冗余消息,通过更改传输消息报文来控制冗余的产生,提出一种基于节点轨迹标签的一致性维护算法,并用此思想改进了洪泛法和谣言算法.新的改进算法在传输的消息报文头部添加已获更新节点的轨迹标签,在发送消息的源头进行冗余判断,极大地减少了冗余消息在已经更新的节点内再次传播的次数,同时,为了减少附加到报文中的节点地址的长度,新算法用 Bloom filter 替代直接存储节点地址标签.通过一致性算法传输理论分析和实验验证,发现改进算法可以大大减少副本冗余传输消息的数目,降低了副本一致性维护代价,增强 P2P 网络的可扩展性,为 P2P 网络开展动态文件更新的新业务提供保障.同时,本文关于副本一致性维护算法的研究还可以用到传感器网络等其他自组织网络的一致性维护中.

References:

[1] Ion S, Robert M, David L, David K, Frans KM, Frank D, Hari B. Chord: A scalable peer-to-peer lookup protocol for internet applications. IEEE/ACM Trans. on Networking, 2003,11(1):17−32.

[2] Ranjita B, Stefan S, Geoffrey V. Replication strategies for highly available peer-to-peer storage systems. Technical Report, CS2002-0726, UCSD, 2002.

[3] Qin L, Pei C, Edith C, Kai L, Scott S. Search and replication in unstructured peer-to-peer networks. In: Proc. of the 16th ACM Int'l Conf. on Supercomputing (ICS 2002). New York: ACM Press, 2002. 84−95.

[4] Wang QB, Dai YF, Tian J, Zhao T, Li XM. An infrastructure for attribute addressable P2P network: Barnet. Journal of Software, 2003,14(8):1481−1488 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/14/1481.htm

[5] Dou W, Wang HM, Jia Y, Zou P. A rumor-spreading analog on unstructured P2P broadcast mechanism. Joural of Computer Research and Development, 2004,41(9):1460−1465 (in Chinese with English abstract). http://crad.ict.ac.cn/papers/2004-9-1460.htm

[6] Datta A, Hauswirth M, Aberer K. Updates in highly unreliable, replicated peer-to-peer systems. In: Proc. of the 23rd Int'l Conf. on Distributed Computing Systems. Washington: IEEE Computer Society, 2003. 76−85.

[7] Zhijun W, Das SK, Kumar M, Huaping S. Update propagation through replica chain in decentralized and unstructured P2P systems. In: Proc. of the 4th Int'l Conf. on Peer-to-Peer Computing. Washington: IEEE Computer Society, 2004. 64−71.

[8] Jiang L, Xiaotao L, Prashant S, Krithi R. Consistency maintenance in peer-to-peer file sharing networks. In: Proc. of the 3rd IEEE Workshop on Internet Applications. Washington: IEEE Computer Society, 2002. 90−94.

[9] Chen X, Ren SS, Wang HN, Zhang XD. SCOPE: Scalable consistency maintenance in structured P2P systems. In: Proc. of the IEEE Infocom 2005. Washington: IEEE Computer Society, 2005. 1502−1513.

[10] Marius P, Aruna S. The cost of application-level broadcast in a fully decentralized peer-to-peer network. In: Proc. of the 7th Int'l Symp. on Computers and Communications. Washington: IEEE Computer Society, 2002. 941−946.

[11] Burton HB. Space/Time trade-offs in hash coding with allowable errors. Communications of the ACM, 1970,13(7):422−426.

[12] Whitaker A, Wetherall D. Forwarding without loops in Icarus. In: Proc. of the IEEE OPENARCH 2002. Washington: IEEE Computer Society, 2002. 63−75.

[13] Xie K, Min YH, Zhang DF, Xie GG, Wen JG. Basket bloom filters for membership queries. In: Proc. of the IEEE Tencon 2005. Washington: IEEE Computer Society, 2005. http://www.tencon2005.org/

[14] Barábasi AL, Albert R. Emergence of scaling in random networks. Science, 1999,286(5439):509−512.

[15] Alberto M, Anukool L, Ibrahim M, John B. BRITE: An approach to universal topology generation. In: Proc. of the MASCOTS 2001. Washington: IEEE Computer Society, 2001. 346−353.

[16] Qin L, Sylvia R, Scott S. Can heterogeneity make Gnutella scalable? In: Proc. of the 1st Int'l Workshop on Peer-to-Peer Systems (IPTPS). London: Springer-Verlag, 2002. 94−103.

[17] Yatin C, Sylvia R, Lee B, Nick L, Scott S. Making Gnutella-like P2P systems scalable. In: Proc. of the ACM SIGCOMM 2003. New York: ACM Press, 2003. 407−418.

附中文参考文献:

[4] 王庆波,代亚非,田敬,赵通,李晓明.基于特征信息定位的 P2P 网络模型:Barnet.软件学报,2003,14(8):1481−1488. http://www.jos.org.cn/1000-9825/14/1481.htm

[5] 窦文,王怀民,贾焰,邹鹏.模拟谣言传播机制的无结构 P2P 网络中广播机制的研究.计算机研究与发展,2004,41(9):1460−1465. http://crad.ict.ac.cn/papers/2004-9-1460.htm

谢鲲(1978－),女,湖南黔阳人,博士生,主要研究领域为可信系统与网络.

谢高岗(1974－),男,博士,副研究员,CCF 高级会员,主要研究领域为下一代互联网.

张大方(1959－),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为可信系统与网络,容错计算.

文吉刚(1978－),男,博士生,主要研究领域为分布式计算与网络.