

基于 P2P 网络环境下的副本一致性维护算法

郭晓梅 李仁发 文吉刚 谢 鲲

(湖南大学计算机与通信学院 长沙 410082)

摘 要 目前副本策略已经被广泛地应用于提高 P2P 系统的性能和可扩展性。随着越来越多的业务的发展,需要文件更新频繁以及网络的高度动态性,使得副本的一致性维护亦变得越来越重要。针对这个问题,研究在高度动态的网络环境下,将结点离线率和副本点与非副本点之间的转换概率作为依据,构建维护责任点分担 owner 责任。并用分层式混合推拉式谣言机制传递更新信息,减少冗余信息。实验证明,新算法在更新过程中能维持高的更新成功率和较低的流量开销,更适应动态网络的扩展。

关键词 一致性维护,对等网络,高度动态性,分层,推/拉技术

中图法分类号 TP393

Replication Consistency Maintenance Algorithm in Unstructured P2P Systems

GUO Xiao-mei LI Ren-fa WEN Ji-gang XIE Kun

(College of Computer and Communication, Hunan University, Changsha 410082, China)

Abstract Replication is a prevalent way to improve the performance and extension as well as to reduce the query responding time in P2P system. Because current network may be dynamic with nodes frequently joins and leaves and objects updated highly for adapting to operation development, maintaining consistency of replications is more challenging in these environment. This paper presented a region push then pull algorithm in highly dynamic networks. It is based on percentage of offline nodes and the transform ratio between replicated nodes and un-replicated nodes to decide the nodes which are responsible of easing the burden of owner. Besides, it can largely reduce the number of the duplicated messages. The experimental results show that the new algorithm has excellent consistency and lower messages overhead especially in dynamic environments, which is better adapted to the extension of dynamic P2P systems.

Keywords Consistency, Peer-to-peer, Highly dynamic, Region, Pull/push

1 引言

P2P 网络可以被大规模地推广和应用,最主要的优势是用户之间可以共享文件,而副本在基于 P2P 的文件共享中占有很重要的地位。目前副本策略已经被广泛地应用于提高 P2P 系统的可伸缩性和性能,缩短响应时间。

以前我们考虑得比较多的是对于一些静态只读和更新无需太频繁的文件共享,而随着社会的发展,人们的需求也越来越广泛,例如在网上开辟的网上拍卖、远程合作、服务指南、在线游戏等业务导致文件更新的频率较高,并且结点加入退出频繁。如何设计有效的一致性维护算法,保障多个副本的一致性,就成为了保证系统正常运行的关键,也是提高 P2P 系统 QoS 和拓展业务的迫切要求。

目前, 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上,而副本的一致性维护并没有得到广泛研究。现有算法的大多数原始思想来自对 C/S 模式缓冲区资源一

致性维护的借鉴。在结构化 P2P 网络中(如 Chord, CAN 等),各结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松散结点组成,没有控制拓扑结构和文件存放,对于数据的可用性和持久性通常只提供粗略的保证。基于洪泛的副本一致性维护算法、基于谣言(Rumor 或 Gossip)的副本一致性维护算法^[1]等实现简单,但洪泛法会在网络中产生大量的冗余信息,而 Rumor 在减少冗余信息的同时副本更新成功率也受到很大的影响。而副本链^[2]的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。此外,动态拉机制^[3]虽然弥补了推技术在动态网以及超过 TTL 限制后的失效性,但是并没有提出适合的 pull 时机和策略。

本文提出的算法是针对现有的无结构 P2P 系统的副本

到稿日期: 2008-02-28 本项目得到湖南省自然科学基金(06JJ2090),国家自然科学基金(60703096, 60703097),湖南省科技攻关项目(05GK3048)资助。

郭晓梅(1982—),女,硕士研究生,研究方向为 P2P 对等网络, E-mail: guoxiaomei1982@126.com; 李仁发(1956—),男,博士,教授, CCF 会员,主要研究领域为嵌入式计算、信息网络与数字媒体; 文吉刚(1978—),男,博士,主要研究领域为分布式计算、P2P 流媒体; 谢 鲲(1978—),女,博士,主要研究领域为可信系统与网络、多网融合。

更新算法冗余消息多、结点出入频繁造成更新信息丢失现象,提出的一种基于分层的混合推/拉技术的一致性维护算法。这里考虑到实际应用中用户的不同需求,更新只传送更新提示而不直接传递更新内容,由副本结点自己决定是否下载更新。在传递更新消息的过程前,借鉴 Scope^[4] 责任分配树的原理,构建副本维护网,再用混合推/拉技术传递更新信息,保证更新成功率,同时减少更新流量代价。

2 基于分层的混合推/拉技术算法

2.1 副本维护网的构建

本算法选择以 Gnutella 为代表的无结构 P2P 网络为环境,所做的工作首先是构建维护副本网的结构。当文件拥有者的信息更新时,通过下述规则发布更新提示,通知其他副本结点,然后根据自身的需要下载最新的版本信息,使得需要及时更新的副本保持一致性。进行消息更新前对副本维护拓扑的构建,如图 1 所示。

1) 文件拥有结点(owner):文件创始者,具有唯一更改文件内容的权利。

2) 所有副本结点分两类:责任点和普通结点。

责任点(responsibility):最早被通知更新信息的结点,建立列表收集与之邻近的副本结点信息。此类结点 TTR 值初始化为最小值,第一时间收到 owner 的更新信息,并且负责将此信息再 push 给下一个 TTL 中拥有副本的邻居结点。

普通结点:责任点的子结点。通常只需被动地接受责任点的 push 更新信息。此类结点 TTR^[5] 初始化为最大值。在当前 TTR 内收不到更新信息时,运用 pull 积极向责任点发送查询信息。

3. 选取责任点:在每个 TTL 值的范围区域内选取一些代表结点(责任点)。根据以下规则设置责任点,步骤如下:

责任点定级:每个副本在建立时都应该存有版本号、owner id 等信息,所以每个副本结点能得到距 owner 的跳数。假设这个副本点与 owner 距离为 d 跳,那么此责任点为 d 级责任点。

每个结点记录自己连通度 n 和拥有副本的邻居结点数 m 。在高动态网络中,考虑到结点的有效性和副本的生命周期,副本点与非副本点之间可能转换,设副本点转换为非副本点的概率为 $1/p$,当这个结点成为责任点的可能性 r 超过 K ,我们就设这个点为责任点:

$$r = (m(1 - 1/p) + (n - m)1/p) / n > K \quad (1)$$

每个责任点返回自身的的信息给 owner。owner 据此建立一个列表,每次更新根据此列表发送更新信息。

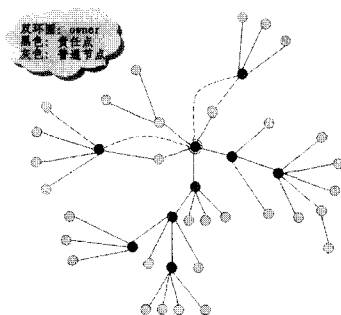


图 1 拥有副本结点的拓扑图

优化列表:原则上,责任点列表只记录与之相邻的结点信息。例如三级责任点 $n3$ 是二级责任点 $n2$ 的子结点, $n2$ 的列表信息就保存了 $n3$ 的信息。 $n2$ 又是一级结点 $n1$ 的子节点, $n1$ 就保存了 $n2$ 的信息。那么 owner 列表就无需保存 $n3$ 与 $n2$ 的信息。

2.2 更新消息的传播机制

当文件被更新,owner 发送更新信息给列表中结点,而各级责任点接收到更新提示后,继续传递给邻居副本点和下级责任点。但在 P2P 网络中,结点出入频繁造成的网络故障或者分裂,使得部分临散副本点错过更新消息,由此引入 pull 补充机制:给每个结点设置动态的 TTR 值,在当前 TTR 值内没有收到更新信息,结点自主发出询问消息。

(1) 动态 TTR 值的选择:

$$TTR = \begin{cases} TTR_{old} + C, & \text{version changed} \\ \frac{TTR_{old}}{D} & \text{otherwise} \end{cases} \quad (2)$$

$C > 0, D = VN_{new} - VN_{old} + a(VN_{new}$ 是最新版本号, VN_{old} 是本结点更新前版本号, $a > 0$)。

如果在当前 TTR 内版本有更新,及时对数据更新,然后对 TTR 值做出修改,即值相应地减少 (TTR/D)。若在当前 TTR 版本无更新, TTR 值就相应地增大 ($TTR + C$),这种自适应 TTR 的变化方式使得离线的结点重新加入网络后可以获取错过的更新信息。

(2) 在 Gnutella 中,网络连通度的变化是作为网络动态变化的标志。所以加入结点平均连接度和自身结点连接度两个因子计算 TTR,减少 pull 带来的侵越性。将式(2)中的 TTR 进一步改进:

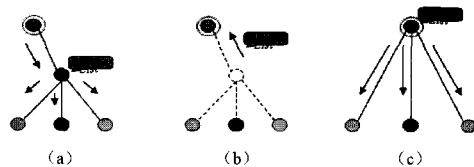
$$TTR = TTR + (1 + \frac{N_{conn} - N_{avgconn}}{N_{avgconn}}) * h \quad (3)$$

利用每个结点的网络联结度与平均网络联结度比较后的大小决定 TTR 的变化取向。当一个结点的联结度小于平均网络联结度时,邻居结点比较少,此结点该积极地搜寻更新信息,相应的 TTR 值就要减少。相反,如果此节点在某时段的网络联结度高,认为此结点比较优势,不急于向外索取更新信息,相应的 TTR 增大。最后, TTR 的有效取值范围为

$$TTR = \max(TTR_{min}, \min(TTR_{max}, TTR))$$

3 副本网优化与维护

当某一区域的责任点离开网络或者失效时,将此级责任点蕴含的子结点信息传至父结点,子结点的阶层提高一级,如图 2 所示。



(a) 双圆环代表 owner,黑色代表责任点,灰色代表普通结点,方框代表结点所保存的列表信息。(b) 一级责任点离开网络或失效。(c) 重新建立的归属关系。

图 2 副本网优化

当结点重新加入网络,首先通信邻近责任点,更新自身版本,重新评估 r 值(见式(1)),判断是否成为责任点。原则上,每个副本点只能有唯一的父结点。为了减少响应时间,选取

级别最高的责任点为父结点,这样避免多个责任点同时向一个子结点发送同样的更新信息。

网络结点频繁加入退出,使得副本点拓扑图在不同时间会有所变化。为了适应这种改变,我们假设每过一段时间重新布置维护网拓扑(见 2.1 节)。如果单纯将某个时间点上 r_{new} 值作为设置责任点的依据没有记忆性,并且使得点维护列表和点与点通信流量代价过大。而基于已有的经验和理论成果,小世界现象是对等网络的一大特征,责任点往往起到簇头的作用,通常这种结点资源利用率高,并且新加入的结点也偏向于连接到已存在的具有较大连接度的结点上^[6],对普通结点贡献大。所以以整个时间段它的在线时间率与新 r 值共同决定这个结点是否继续“连任”更加符合网络特征。而一些“表现”极好的普通结点也可由此晋升为新的责任点。

4 仿真实验

4.1 仿真环境及参数设置

为了衡量 RTPP 算法的更新成功率和更新信息流量代价,通过仿真的方法模拟了一个高动态对等网络环境,分别对洪泛算法和 RTPP 算法进行评价。表 1 列出模拟高动态非结构化对等网络环境的主要参数。

表 1 动态对等网络环境参数

参数	描述	缺省值
Lsim	仿真时间长度	10h
Rf	最高结点离线率	50%
If	相继离线平均时间	5min
Df	平均离线时间	2h

我们设置的整个网络由 500 个结点和 5000 个文件对象组成,资源按 20/80 规则分布,即 20%的结点拥有 80%的网络资源。据实验证明,95%的结点很少在 7 跳之外,所以将 TTL 设为 8。表 2 列出了算法主要模拟参数。

表 2 算法模拟参数

参数	描述	缺省值
TTRmax	TTR 最大值	1h
TTRmin	TTR 最小值	5min
Navgconn	结点平均联通度	4
p	1/副本转换为非副本点概率	3
K	设置责任点阈值	1/2
C	动态 TTR 参数,公式 1	10min
a	动态 TTR 参数,公式 2	0.5
h	动态 TTR 参数,公式 3	10

4.2 仿真结果

我们使用两个不同的评价指标评估 RTPP 算法性能:丢失更新率 Mumr(missed update messages ratio)和更新信息流量代价 Ofmu(overhead of messages per update)。丢失更新率衡量算法在结点离线率 5%~50%时更新消息丢失的情况。如图 3 所示,横坐标是最高结点离线率,纵坐标是丢失更新率。图中给出了 Flooding 和 RTPP 算法在结点不同离线率时更新消息丢失的对比数据。在结点状态相对稳定时,Flooding 和 RTPP 的更新成功率都很高。随着结点离线率的增高,Flooding 丢失的更新信息数目直线上升,相比而言,

RTPP 对网络高动态变化的适应能力比较强,信息丢失比率一直保持在比较理想的状态。

更新信息流量代价也是副本更新消息传输开销,图 4 横坐标是网络规模,纵坐标是更新副本开销,表示了在不同网络规模(200~2000)Flooding 和 RTPP 消耗的副本更新信息传输量。Flooding 随着网络规模不断扩大,传输开销增长很快,冗余信息量大。RTPP 除了考虑结点连接度的状况,还考虑了含有邻近副本点的比例,选取责任点传递更新信息,避免了 Flooding 的盲目性。

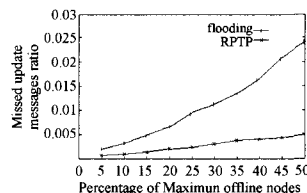


图 3 丢失更新率

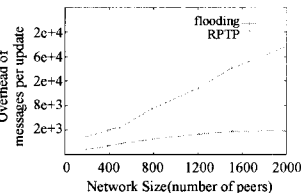


图 4 更新信息流量代价

结束语 本文主要讨论非结构化 P2P 动态网络资源副本一致性维护算法。在高动态网络环境下,构建分层的副本维护网,结合推拉技术保证副本的一致性。实验表明,有效责任点的引入使得更新消息传递避免了传统 Flooding 机制的盲目性,大量减少了冗余信息的开销,减轻了网络负载和避免了单点瓶颈的问题。而将推拉技术相结合,在网络结点出入频繁的情况下,使得孤单结点和重新加入的结点可以自主地搜索更新信息,维持高的副本更新率。

参 考 文 献

- [1] Datta A, Hauswirth M, Aberer D. Updates in highly unreliable, replicated p2p systems[C]// Proc. of 23th IEEE International Conference on Distributed Computing Systems, Providence Rhode island, USA, 2003: 76-85
- [2] Wang Z, Kumar M. Update propagation through replica chain in decentralized and unstructured p2p systems[C]// Proc. of the 4th International Conference on p2p Computing, 2004. 1730 Massachusetts Ave., NW Washington, DC USA: IEEE Computer Society, Aug. 2004: 64-71
- [3] Duvvuri V, Shenoy P, Tewari R. Adaptive Leases: A Strong Consistency Mechanism for the World Wide Web[J]. IEEE Transactions on Knowledge and Data Engineering, 2003, 15(5): 1266-1276
- [4] Chen Xin, Ren Shansi, Wang Haining, et al. SCOPE: scalable consistency maintenance in structured P2P systems[C]// Proc. of IEEE INFOCOM 2005 Conference, Miami, Florida, March 2005, 3: 1502-1513
- [5] Lan Jiang, Liu Xiaotao, Shenoy P, et al. Consistency Maintenance in Peer-to-Peer File Sharing Network[C]// Proc. of The Third IEEE Workshop on Internet Applications (WIAPP'03), June 2003: 90-94
- [6] Wang Xiaofan, Chen Guanrong. Complex networks: small world, scale-free and beyond [J]. IEEE Circuits and Systems Magazine, 2003, 3(1): 6-20