Research on Keeping Replica Consistency in Unstructured P2P Network

GUAN Zhitao, WU Kehe, HE Jietao

Department of Computer, North China Electric Power University, Beijing, China

e-mail: *guanzhitao@126.com*

Abstract: Aiming at the problem that it hard to keeping replica data consistency in unstructured P2P network, a novel data replica consistency update strategy is proposed. The replica consistency scheme consisted of two parts on the basis of the current popular super-peer structure of p2p network. First part, in super peer layer, the replica consistency is maintained by using Inter-Domain Version Vector. Second part, in the domain of any super peer, the consistency is maintained by using Inner-Domain Version Vector. This double-layer consistency maintenance strategy can not only keep the replicas be consistent, but also reduce the bandwidth caused by the update operation as well. The simulation results indicate that DVV shows better performance than the compared strategies in the current unstructured p2p network.

Keywords: peer-to-peer; super-peer structure; replica consistency

非结构化 P2P 网络中的副本一致性策略研究

关志涛，吴克河，何杰涛

华北电力大学计算机系，北京，中国，102206

e-mail: guanzhitao@126.com

【摘　要】针对 P2P 网络中数据副本一致性维护困难的问题，提出了一种新的副本一致性维护方法。该策略利用当前流行的超节点结构，将副本更新操作分为两个层次。在超节点层使用域间版本向量保证更新一致性；在超节点域内，通过超节点控制域内副本更新。这种分层的一致性维护策略既保证了副本一致性，又减少了因更新操作而引起的带宽消耗。实验表明，和已有流行策略相比，DVV 在进行一致性维护时消耗的消息包数更少，且副本更新速度明显加快，在当前非结构化网络中具有更好的性能。

【关键词】对等网络；超节点结构；副本一致性

1 引言

在 P2P 环境中使用副本技术可极大地提高数据的可靠性与访问效率。即使节点失效，数据仍可从备份节点读取，从而使数据更加可靠。通过有选择地访问延迟较小的节点，也提高了数据的访问效率。但如何保证数据副本的一致性是需要解决的问题。Iamnitchi 利用 gossip 协议[1]将资源索引信息散布给同一簇的节点，实现资源索引信息的各份和动态更新[2]。Datta 提出基于 rumor spreading[3]的更新传播算法，每个节点保存所有副本节点的子集作为其负责的节点。当一个副本节点更新时，更新被传播给其负责的节点，再由这些节点转发给它们负责的节点。算法开销较大，并仅提供高概率弱一致性的更新[4]。Wang 提出通过副本链进行更新传播的算法(Update Propagation Through Replica Chain，简称 UPTReC)[5]。在 UPTReC 中，对于每个资源，复制此资源的所有节点相互连接形成一条分布式的逻辑副本链。当一个副本节点启动一次更新时，它将通过副本链把更新信息传播到所有副本节点。这些资源副本一致性更新算法只能提供高概率的弱一致性更新。

乐观复制（Optimistic Replication）是一种近来出现的一致性维护方法。乐观复制不要求各数据副本时刻一致，因此适合在广域网及移动环境下使用，节点高度自治，可独立处理查询和更新操作，数据副本发生冲突时可使用冲突解决策略使数据恢复一致。乐观复制已成功

* 华北电力大学校内基金支持（200822042）

应用在异步协作系统 Bayou、点对点复制文件系统 Ficus 等系统上。乐观复制方法有因果历史（Causal History）[6]，版本向量（Version Vector）[7]等多种实现形式。

本文提出了一种低开销、层次式的副本一致性策略 DVV(Domain Version Vector)。将副本一致性维护操作分为域间和域内两个层次。在超节点层使用域间版本向量保证更新一致性；在超节点域内，通过超节点控制域内副本更新。这种分层的一致性维护策略既保证了副本一致性，又减少了因更新操作而引起的带宽消耗。

2 DVV

在 P2P 环境中使用副本技术可极大地提高数据的可靠性与访问效率。即使节点失效，数据仍可从备份节点读取，从而使数据更加可靠。通过有选择地访问延迟较小的节点，也提高了数据的访问效率。但如何保证数据副本的一致性是需要解决的问题。若文件为只读属性，则不存在副本一致性维护问题；若文件为可写属性，那么当某个文件副本更新时，其余副本也需要更新以保证副本的一致性。

版本向量(Version Vector,VV)是一种有效的副本副本一致性维护策略[7]。在集中式网络环境下，版本向量方法可以起到很好的作用。但是，在 P2P 环境中，同一个文件的副本数量可能很大，而且通常分布很广，此时使用版本向量进行一致性更新管理，向量会变得过于庞大而不易维护。我们基于版本向量方法，对版本向量进行改进，提出一种基于域版本向量(Domain Version Vector, DVV)的分层副本一致性维护策略，将副本一致性维护操作分为域内和域间两个层次，在域内和域间分别应用版本向量进行一致性维护（分别称之为域内版本向量和域间版本向量），在超节点所属域内应用版本向量模式进行域内一致性维护，在超节点层进行域间一致性维护，从而达到有效维护整个网络副本一致性的目标。

2.1 域内一致性维护

2.1.1 域内版本向量

域内副本的一致性维护方法是基于域版本向量的。域版本向量的结构与版本向量类似，最主要的区别在于域版本向量比版本向量多了最近更新标记位。表 1 所示为副本 *X* 所对应域内版本向量的结构。表中每列包含两项，上面一项是节点项，其内容为节点 ID；下面一项是副本 *X* 在该节点处的更新计数值。表的最后一列是一个布尔标记，表示该副本最近一次更新是由哪个节点发起的。*Flag* 为 1 表示最近一次更新由超节点发起，为 0 则表示更新是由普通节点发起。发起更新操作的节点不同，其操作也不相同。

2.1.2 域内副本更新策略

域内副本更新基于域内版本向量进行的。首先给出主副本的概念。

Table 1. The structure of inner-domain VV
表 1 域内版本向量(DVV)结构

节点 P1	节点 P2	…	节点 Pn	最近更新标记 *Flag*
$Count_{P1}(X)$	$Count_{P2}(X)$	…	$Count_{Pn}(X)$	0 或 1

图 1 描述了使用域内版本向量处理不同节点间版本更新合并的一个例子。

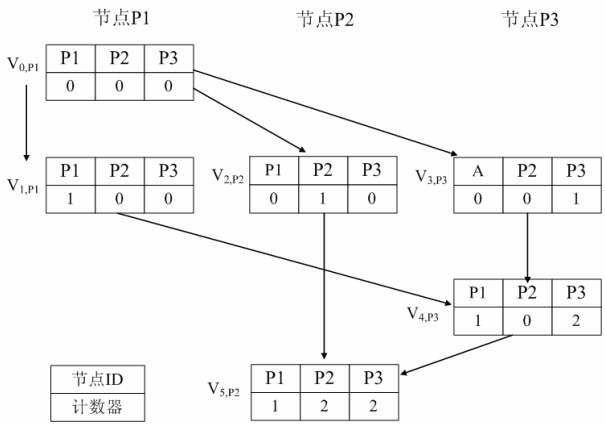

Figure 1. Inner-domain version vector
图 1. 域内版本向量图示

定义 1: 主副本(Primary replica)

节点 P 为域 D 内的超节点，若对象 X 在域 D 内有多个副本，且 X 在超节点 P 上存在副本 R，称 R 为对象 X 在域 D 内的主副本。

当对象 X 有更新操作发生时，更新扩散策略分为两种情况：

A. 超节点对 *X* 发起更新操作，即主副本发生更新的情况。这种情况发生的前提有两个：(1). 超节点自身更新主副本；(2). 其他域对 X 进行了更新操作，并将更新信息扩散所有拥有该副本的域。对于这种情况，其更新操作包括两个步骤：域内更新和域间更新。其中域

978-1-935068-06-8 © 2009 SciRes.

内更新；域间更新操作的具体方法在下一节介绍。

B. 普通节点发起更新操作。该节点将更新信息传递给域内超节点，接下来的步骤同步骤 A。

需要说明的是，超节点的存储能力是有限的，所以在超节点为每一个对象都维护一个主副本是不现实的。因此，超节点处主副本使用 LRU 策略进行淘汰。被淘汰者进入休眠队列，保留索引，只进行域内副本更新，不进行域间更新。当域间更新发生时，可再次从休眠队列中被唤醒。

2.2 域间一致性维护

域间副本的一致性维护策略也是基于版本向量方法，下面给出具体策略描述。

2.2.1 域间版本向量

使用域间版本向量来维护不同域间的同一个文件副本的一致性。如表 2 所示，域间版本向量(Inter-Domain Version Vector, IDVV)的结构与域版本向量(DVV)很相似，不同的是域间版本向量没有 Flag 标志位。因为二者的操作基本一致，此处不再对 IDVV 做详细说明。

Table 2. The structure of inter-domain VV
表 2 域间版本向量(DVV)结构

节点 *P1*	节点 *P2*	…	节点 *Pn*
$Count_{P1}(X)$	$Count_{P2}(X)$	…	$Count_{Pn}(X)$

2.2.2 域间副本更新策略

当对象 *X* 有更新操作发生时，更新扩散策略分为两种情况：

A. 超节点 *P* 对 *X* 发起更新操作，即主副本发生更新的情况。节点 *P* 将 *X* 更新后，修改域间版本向量，将自身对应的向量位的计数值加 1。之后通过谣言传播机制，将 *X* 的更新信息及其对应的域间版本向量打包，扩散其他拥有该副本的超节点。其他超节点收到该更新后，启动域内副本更新机制，保证该域内副本更新一致性，具体过程如前文所述。

B. 普通节点发起更新操作。该节点将更新信息传递给域内超节点，接下来的步骤同步骤 A。

图 2 给出了一个域间一致性维护与域内一致性维护如何结合的例子。在超节点 A,B 和 C 之间使用域间一致性维护操作，之后每个超节点在各自域启动域内一致性维护操作，最终达到全局一致性。

3 仿真实验

3.1 实验环境

本文选取 P2P 文件共享系统作为模拟设定场景，并以 Gnutella 的动态查询方法[8]作为底层查询协议，超节点层拓扑依照当今 Gnutella 的结构进行组织[9]，共包含 10240 个超节点，超节点间连接度分布符合幂律分布，如图 3 所示。超节点层拓扑由 BRITE[10]生成。每个超节点的叶节点数为[20,30]区间内一随机数。系统中包含若干类文档，每类文档用一个关键字进行标识。初始化时，每类文档拥有相同数量的副本，均匀地随机分布在各节点上。仿真程序由 Java 编写，查询模拟是按轮(run)进行的。每轮随机选择一节点发起查询。

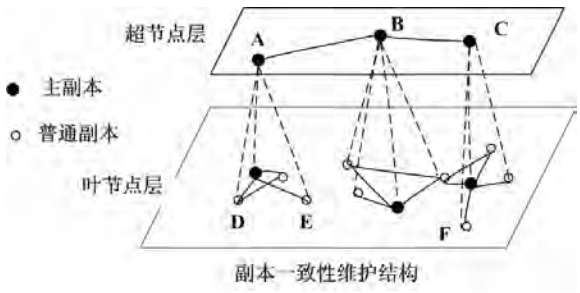

Figure 2. Hierarchical structure replica consistency maintenance
图 2 层次式副本一致性维护图示

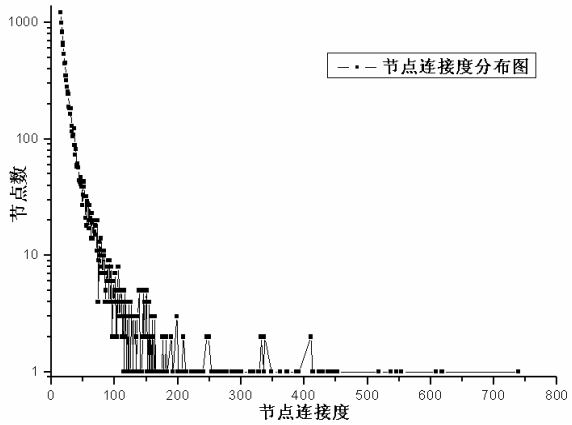

Figure 3. Peer connection degree distribution of the simulation network

图 3 仿真网络节点连接度分布图

3.2 实验结果及分析

在发起相同副本更新操作的前提下，从以下两个方面比较我们提出的基于域版本向量 DVV(Domain Version Vector)的一致性策略与基于版本向量 VV(Version Vector)的一致性策略的性能：

1、副本更新后，用于维护副本一致性的消息数量；

2、副本更新后，副本版本更新的扩散速度，即单位时间内的更新覆盖率。

实验共运行 50 轮。每轮随机选取节点发起副本更新操作。

图 4 所示为分别使用域版本向量(DVV)一致性策略和版本向量(VV)策略进行一致性维护时消耗的带宽对比。其中 DVV-Average 和 VV-Average 分别表示在 50 轮实验中 DVV 策略和 VV 策略实验结果的均值。可以看出，DVV 策略消耗的消息包更少，明显优于 VV 策略。原因在于 VV 策略没有考虑超节点因素，在整个网络内对一个文件副本只维护一个版本向量，进行副本更新时就很繁琐，而且容易产生大量的冗余消息。而 DVV 采用了分层的副本更新策略，将副本更新操作分别在超节点层和超节点域内两个层次进行，不同域内的普通节点不需直接通信，自然就降低了因副本更新而引起的网络流量。

图 5 所示 DVV 策略和 VV 策略的副本更新速度对比。横轴表示实验运行轮数，纵轴表示在每一轮实验中副本更新所覆盖到的节点百分比。同样，DVV-Average 和 VV-Average 分别表示在 50 轮实验中 DVV 策略和 VV 策略实验结果的均值。可以看出，DVV 策略的更新速度快于 VV 策略，因为基本每轮实验 DVV 策略的覆盖到的节点百分比都高于 VV 策略，而且 DVV-Average(0.801) 也明显高于 VV-Average(0.563)。其原因同样在于 DVV 采用了层次的副本更新方法，副本更新首先在超节点层散播，然后超节点在自己所属域内进行副本更新，层次清晰，副本更新速度也因此而加快。

4 结论

本文提出一种低开销、层次式的副本一致性策略 DVV。将副本一致性维护操作分为域间和域内两层。在超节点层使用域间版本向量保证更新一致性；在超节点域内，通过超节点控制域内副本更新。这种分层的一致性维护策略既保证了副本一致性，又减少了因更新而引起的带宽消耗。实验表明，DVV 是有效的。

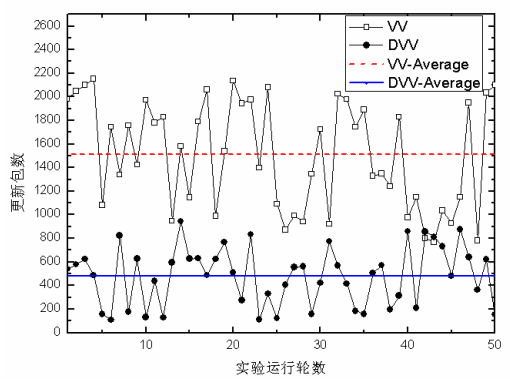

Figure 4. Comparison of the bandwidth for replica consistency maintenance

图 4. 因一致性维护而消耗的带宽对比

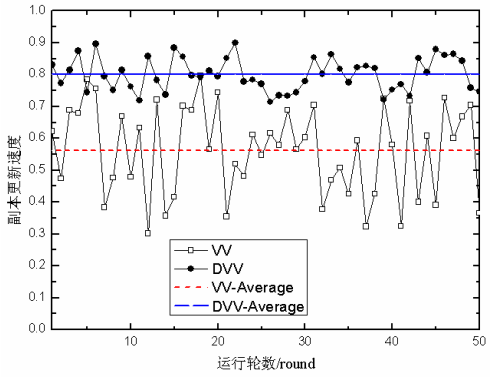

Figure 5. Comparison of the replica proliferation rate

图 5. 副本更新速度对比

References (参考文献)

[1] Kermarrec A M, Massoulie L, Ganesh A J. Probabilistic reliable dissemination in large-scale systems[J]. IEEE Transactions on Parallel and Distributed Systems. 2003, 14(3): 248-258.

[2] Iamnitchi A, Ripeanu M, Foster I. Locating Data in (Small-World?) Peer-to-Peer Scientific Collaborations[A]. 1st International Workshop on Peer-to-Peer Systems (IPTPS'02)[C]: 2002.

[3] Karp R, Schindelhauer C, Shenker S, et al. Randomized rumor spreading[A]. IEEE Symposium on Foundations of Computer Science[C]: 2000. 565–574.

[4] Datta A, Hauswirth M, Aberer K. Updates in highly unreliable, replicated peer-to-peer systems[A]. Distributed Computing Systems, 2003. Proceedings. 23rd International Conference on[C]: 2003. 76-85.

[5] Wang Z, Das S K, Kumar M, et al. Update propagation through replica chain in decentralized and unstructured P2P systems[A]. Peer-to-Peer Computing, 2004. Proceedings. Proceedings. Fourth International Conference on[C]: 2004. 64-71.

[6] Kang B B, Wilensky R, Kubiatowicz J. The hash history approach for reconciling mutual inconsistency[J]. Distributed Computing Systems, 2003. Proceedings. 23rd International Conference on. 2003: 670-677.

[7] Hasegawa K, Higaki H, Takizawa M. Object Replication Using Version Vector[A]. Proc. of the 6th IEEE Int'l Conf. on Parallel and Distributed Systems(ICPADS-98)[C]: 1998. 147-154.

978-1-935068-06-8 © 2009 SciRes.

[8] Fisk A, Gnutella Dynamic Query Protocol v0. 1, in, Editor^Editors. 2003, Gnutella Developer's Forum. p.
[9] Stutzbach D, Rejaie R. Characterizing the two-tier gnutella topology[A]. Proceedings of the 2005 ACM SIGMETRICS international conference on Measurement and modeling of computer systems[C]: 2005. 402-403.
[10] Medina A, Matta I, Byers J, BRITE: A Flexible Generator of Internet Topologies, in, Editor^Editors. 2000. p.