第 35 卷 第 7 期 计 算 机 工 程 2009 年 4 月
Vol.35 No.7 Computer Engineering April 2009
·软件技术与数据库· 文章编号：1000—3428(2009)07—0076—02 文献标识码：A 中图分类号：TP311.52

基于存储资源主题分组的 P2P 存储系统

蔡 亮[1]，黄 浩[2]
(1. 湖北汽车工业学院信息管理系，十堰 442002；2. 湖南大学计算机与通信学院，长沙 410082)

摘 要：利用 P2P 方法建立一个 P2P 存储系统，对参与节点按资源主题数和预测网络距离进行分组，形成由超节点维护的覆盖网络，实现对节点的有效管理，同时使用节点多重命名机制提高系统存储效率，并在仿真实验基础上，对该存储系统的性能进行验证。
关键词：P2P 存储系统；主题覆盖网络；分组；副本

P2P Storage System Based on Storage Resource Topic Grouping

CAI Liang[1], HUANG Hao[2]
(1. Dept. of Information Management, Hubei Automotive Industries Institute, Shiyan 442002;
2. College of Computer & Communication, Hunan University, Changsha 410082)

【Abstract】 By using the way of P2P, a P2P storage system is built. The nodes involved in the system are grouped by the number of the subject of resources and the predicted network distance. The overlay network maintained by super nodes is formed, which implements effective management of nodes. Moreover, by using the multiplex naming mechanism of nodes, the efficiency of data storage in system is promoted. The performance of this storage system is testified on the basis of simulation experiment.
【Key words】 P2P storage system; topic overlay network; grouping; replication

P2P 存储系统的目标是组织系统中的节点来存储数据，为提高系统存储数据的有效性和提高节点对存储资源的有效管理，本文提出一种按 P2P 网络存储资源主题数目及预测网络距离对节点进行分组管理，并最终形成一种覆盖分层的混合式 P2P 存储系统的方法。

1 相关工作

目前，基于 P2P 的存储系统节点管理主要分为 2 种方式：中央集中式(centrailized)和非中央集中式(decentrailized)。典型的集中式系统 Napster 利用中央服务器负责目录管理的服务常受服务器的限制，存在诸多不足，如单点崩溃导致系统拓扑结构被破坏，节点存储数据可用性不高，节点存储数据可靠性下降等问题；典型的非集中式系统 Gnutella 和 Freenet 由于没有中央服务器，查询数据时以 flooding 的方式将消息传播到网络上，存在占用大量系统带宽、消息泛滥和系统可扩展性差等问题。

2 节点分组算法与数据存储机制

2.1 基于存储资源主题和预测网络距离分组算法

在实际网络中，每个节点包含的存储资源都有特定的主题。设 P2P 网络中所有存储资源的主题集合为 $T=\{T_i \mid 1\leqslant i\leqslant n, n\in N\}$，$n$ 为最大可能主题个数；每个主题 T_i 中包含 $P=\{P_j \mid 1\leqslant j\leqslant m, m\in N\}$ 个子主题，m 为 T_i 中最大可能子主题个数。$|T|$ 为网络存储资源主题个数，$|P_j|$ 为 T_i 主题中所含子主题个数。以 $|T|$ 为网络分组个数基准，$|P_j|$ 为各分组内超节点个数基准。

为提高 P2P 存储系统存储空间分配率及存储数据时路由效率，使网络节点更均匀地分布于各分组之中，采用预测的网络距离为半径划圆的方法进行节点分组。具体方法是：在大规模网络中随机取一个节点 k，判断 k 与其相邻节点间网络数据传输时延。假设其有 n 个相邻节点，节点 k 与这 n 个相邻节点间网络可用带宽分别为 $BIT_1, BIT_2, \cdots, BIT_n$，假设传输数据块大小为 $DATA_k$ (单位为 Kb)，则可测得节点 k 与这 n 个相邻节点间网络数据传输时延分别为 $TIME_i = DATA_k / BIT_i$，n=1,2,⋯,n，再取其网络数据传输时延平均值 $D_1 = (\sum_{i=1}^{n} TIME_i)/n = (\sum_{i=1}^{n} DATA_k / BIT_i)/n$，$n$=1,2,⋯,$n$，依此类推，可取得 m 个随机节点的网络数据传输时延的平均值 $D_1, D_2, \cdots, D_m$，假定网络距离的预测值为将所得各采样节点网络数据传输时延的平均值再取平均值，即预测的网络距离 $dis = (\sum_{i=1}^{m} D_i)/m$。得到预测网络距离后，从初始网络中随机选取一个节点，以该节点为圆心，预测网络距离为半径内的所有节点都划入一个分组，再将获得分组后的节点从初始网络中移除。以此方法循环划分 $|T|$ 次，可将初始网络中节点划为 $|T|$ 个分组。对于每个分组 T_i，在分组内选取 $|P_j|$ 个节点作为超节点(考虑到大规模网络的复杂性，主要选取在线时间长的节点)，超节点与其附近普通节点连接成 chord[1]结构，分组内各超节点也连接为 chord 结构。为保证各分组之间的连通性，从各分组内取一个超节点彼此连接为网状结构，如图 1 所示，形成一个 3 层覆盖网络。为保持网络拓扑结构的完整，为分组中每个超节点选取一个其邻居节点作为其备份节点，对连接各分组的超节点，选取其 2 个邻居节点作为其备份节点；每个超节点和其备份节点都保存各自 2 个前继节点和后继节点的位置信息，以保证出现节点失效时拓扑环路不断开。

作者简介：蔡 亮(1979－)，男，助教、硕士研究生，主研方向：分布式存储系统；黄 浩，硕士研究生
收稿日期：2008-10-10 **E-mail：**qycailiang@163.com

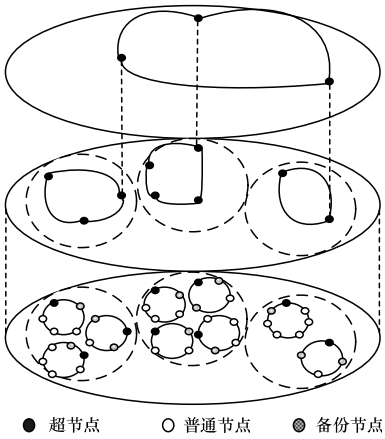

图 1　3 层覆盖网络

2.2　节点命名与数据存储机制

由于采用了对 P2P 网络节点分组最终形成覆盖网络的机制，因此各分组内节点之间网络距离相近。为更充分利用带宽，降低存储数据时路由开销，采用对系统节点按存储资源主题的多重命名机制，即对系统的$|T|$个分组，先将每个分组命名为$T_i(1\leqslant i\leqslant|T|)$，再将组内各超节点按其子主题个数分别命名为$T_{i,j}(1\leqslant i\leqslant|T|,1\leqslant j\leqslant|P_j|)$，最后与各超节点相连的普通节点按 chord[1]规则，利用 Hash 函数命名其标志符，如图 2 所示。

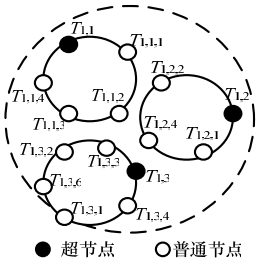

图 2　节点命名

由于各组组名不同，各组内节点对应标志符不同，因此系统中每个节点的名字都是唯一的。为降低数据查找存储位置的系统开销，第 3 层超节点的位置信息公开，为避免其超节点负担过重，其上不存储数据，只负责维护其所连接节点的信息表。在数据存储查询时，只要知道所存储数据的主题，根据主题的相似性及第 3 层超节点所维护的信息，即可确定其要存储的分组及存储节点信息的第 2 层子主题超节点，最后通过查询子主题超节点所维护的普通节点信息表，将存储数据文件名按相同的 Hash 函数映射后，总可找到标志符最相近的节点，将数据存储于该节点上。当有节点需加入系统时，先计算其与各分组超节点之间的网络距离，取该距离平均值最小的分组加入，再连接到该分组中距离最近的超节点环上，将其信息注册于该超节点中；当有节点失效或退出系统时，直接删除该节点信息，再将其前继和后继节点连接上即可。当超节点失效时，立刻用备份节点取代其位置。

3　数据备份及存储空间回收机制

3.1　数据备份机制

数据备份的目的是提高 P2P 存储系统数据的可用性和可靠性，使系统不因某些节点失效而丢失其存储的数据。文献[2]统计分析表明，在特定分组子网中，数据主题查询过程的重复率很高，这是因为网络存在一些存储着热点主题的高访问率节点。这些高访问率热点数据主题的备份，对整个系统可用性和可靠性的提高起着重要作用。因此，本文按主题访问率的高低采用不同的数据备份策略。对于访问率较低的主题数据，采用组内备份的方式，具体方法是：需备份数据节点随机均匀地选择 n 个(该参数为创建副本的数目，需用户预先设定)组内节点，在这些节点上创建该数据的副本，并将此信息写入与其相连的超节点信息表中。对访问率高的主题数据，采用组内备份和组间备份相结合的方式，即对访问率高的数据在组内节点备份的同时，也将该数据有选择地备份到其他的分组节点中，再将其备份信息写入与其相连的超节点信息表。由于存储网络采用的是覆盖分层的超节点管理拓扑结构，因此当有节点存储数据失效时，系统可通过查询超节点信息表方式在组内或组间快速找到所需备份数据信息。其中，组间备份方式较组内备份方式容灾性更强，更适合于大规模存储系统中重要数据信息的存储，从而有效保证访问率高数据的完全可修复性，在不给系统增加过重负担的情况下，提高系统性能。

3.2　存储空间回收

当有节点失效或永久退出系统时，系统会要求相关节点删除其保存的该节点信息，但在删除数据过程中可能存在部分节点暂时不在线，而导致这些节点上的相关信息没有被彻底删除，这将产生存储垃圾，造成存储空间的浪费。因此，在每个节点为每个存储数据附加一个最后访问时间属性的记录域，对每个最后访问期限超过一定范围的数据，节点有权删除。随着时间的流逝，未被访问的数据将逐渐被删除，节约了系统资源。

4　性能分析

利用 Intel P4 1.6 GB 处理器和内存 512 MB 的计算机，在 Linux Redhat 9.0 环境下，使用 NS-2 网络仿真器实现对实际网络状况的模拟，并通过 Otcl 和 C++程序设计分别实现 Chord 系统模型和本文算法。

考虑到存储系统的性能要求，将确定存储数据位置所需的路由跳数和节点加入与退出系统的平均消息数作为衡量系统性能的重要指标。为测试其性能，设定初始分组数$|T|$的值为 32。具体数据存储操作设定为空间范围数据存储查询，随机选取一个分组间超节点作为数据存储查询的启动节点，性能监视器收集跳数衡量参数。针对不同的查询范围及节点数进行多次实验，分别记录各次的结果，最终取平均值。图 3 给出了本文方法与 Chord 系统模型在不同的节点数时的数据存储查询跳数对比结果，实验结果显示在节点数目较少时，2 种方法效果基本相同，但随着节点数目的增加，本文方法在减少跳数上有较好效果。

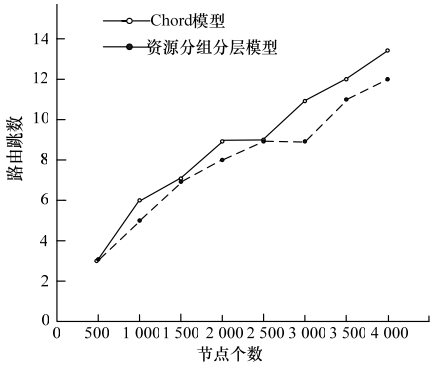

图 3　数据存储查询跳数对比

(下转第 81 页)