·计算机工程与应用·

基于大规模分布式副本定位的分级索引压缩机制

陈建英[1,2]，刘心松[1]

(1. 电子科技大学计算机科学与工程学院 成都 610054; 2. 西南民族大学计算机科学与技术学院 成都 610041)

【摘要】针对超级节点索引方式下的大规模分布式系统，提出一种用于副本定位的资源索引分级压缩机制。该机制把超级节点所辖分级网络中上层节点的有序子节点集映射到一个位串向量，进而通过自下而上的索引发布和索引在上级节点的汇聚实现冗余副本记录数的压缩，副本定位则通过逆向的位串查询实现。实验表明，该机制可达到较高的记录压缩比，并在一定程度上提高副本定位效率。

关 键 词 汇聚； 分级压缩； 大规模分布式系统； 发布； 副本定位

中图分类号 TP393; TP311 **文献标识码** A doi:10.3969/j.issn.1001-0548.2011.04.016

Novel Hierarchical Index Compression Mechanism Based on Large Scale Distributed Replica Location

CHEN Jian-ying[1,2] and LIU Xin-song[1]

(1. College of Computer Science & Engineering, University of Electronic Science and Technology of China Chengdu 610054;
2. College of Computer Science & Technology, Southwest University of Nationalities Chengdu 610041)

Abstract Aiming at super-node indexed large scale distributed system (LSDS), a kind of novel hierarchical index compression mechanism is put forward for replica location. It is implemented by mapping all sub nodes of higher-up node to corresponding bit string at first, then compressing record number of redundant replica index by bottom-up index publish and index aggregating on higher-up node. Accordingly, replicas in system can be located by query on bit string conversely. The experimental results indicate that this mechanism achieves high record compression ratio and has some active impacts on efficiency of replica location in LSDS.

Key words aggregating; hierarchical compression; large scale distributed system; publish; replica

随着网络规模的不断扩大，共享资源与日俱增，共享资源索引空间膨胀问题凸显，特别是通过冗余副本和冗余资源索引提高系统鲁棒性、可扩展性和服务性能的大规模分布式系统(replicas-based large scale distributed system，R-LSDS)中，平均副本冗余度为r的资源索引记录总数高达$O(r^2)$。因此，合理组织和有效压缩海量资源索引是R-LSDS副本定位不可忽视的一项重要内容。

海量资源索引的组织与全局索引方式相关。早期集中的索引方式[1]已不能满足当前网络规模进一步扩大的需求，全分布的索引方式[2]也不适宜高性能的大规模网络应用服务，因此，兼有集中式和全分布式索引优点的超级节点索引方式[3-6]倍受推崇。资源索引的压缩则与压缩目标相关。现有压缩方法和技术基本上都是为了解决数据存储和传输开销问题，包括Bloom Filter压缩法[7]、属性类型分类压缩法[8]、符号表压缩法[9]、海量关系压缩拆分技术[10]等与资源索引压缩最为接近的数据表压缩法，虽然这些方法均可有效压缩冗余字段值，并因此获得查询效率收益，但这种基于单表的压缩方式并不适用于超级节点索引方式下的分级二元资源索引〈资源名称，位置〉，全码特性、按资源名称定位的特殊查询方式，以及上下级索引间的关联关系需要与之相适应的压缩机制。到目前为止，未见与之相关的文献或报道。

鉴于此，本文基于超级节点索引方式提出一种全新的资源索引分级压缩机制，基于物理临近构建了区/站/节点/用户4级分层和区间对等的混合覆盖

收稿日期：2010－10－03；修回日期：2010－12－07
基金项目：四川省应用基础研究项目(2008JY0070-2)
作者简介：陈建英(1970－)，女，博士生，副教授，主要从事分布式并行数据库系统、数字有机体数据库系统方面的研究.

网络结构，并在由区/站/节点3级构成的分级索引网络中，把索引节点的有序子节点集映射为相应长度的位串向量，进而制定映射规则将所有同名副本记录压缩为一条，副本定位则根据映射规则逆向查询位串实现。实验表明，该压缩机制可有效缩减资源索引记录数，达到较高的记录压缩比，并对副本定位效率的提高起到一定的作用。

1　系统覆盖网模型

有较多文献对大规模分布式系统(large scale distributed system，LSDS)覆盖网络模型进行了深入的研究，普遍采用超级节点层(super-node layer network，SLN)对等、超级节点所辖子网(local hierarchy network，LHN)分层的覆盖网络构建模式。在LSDS中，若记M为平均子节点容量，p为平均节点负载率，k为LHN层数，系统规模为n，则SLN中的节点数仅为$n/(Mp)^{k-1}$。另外，考虑到基础网络对覆盖网络路由性能的影响[11]，本文基于物理网络把LSDS组织为如图1所示的混合覆盖网络结构。其中，区/站/节点3级是服务器构成的相对稳定的资源索引网络，区级和普通用户级节点分别构成局部对等网络，用户级节点不参与索引管理。

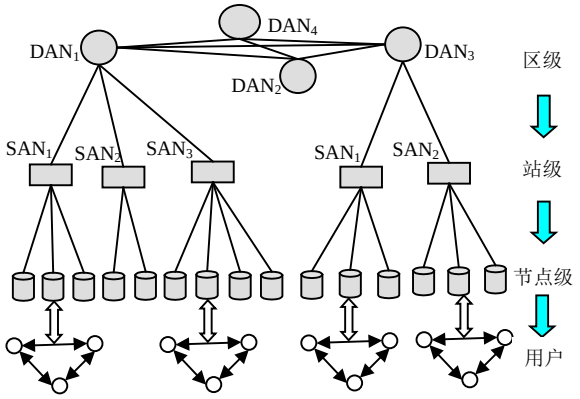


图1　LSDS覆盖网络模型

区/站/节点3级索引节点按功能区分如下：

节点级节点也是共享资源存储节点(resources storage node，RSN)，本地实际存储共享资源并负责管理用户级节点和本地资源索引的服务器。通常把同一局域网或物理临近的多个RSN组织为一个“站”并生成相应的站代理节点(site agent node，SAN)。

站代理节点即站级节点，专门配置或在本站RSN中选取综合性能(在线时间、网络带宽、节点处理能力等)相对较好的一个或多个节点，负责站内资源索引的管理与维护。通常将物理位置相邻或有着密切联系的多个站组织为一个“区”，并生成相应的区代理节点(district agent node，DAN)。

区代理节点即区级节点，专门配置或在本区RSN中选取综合性能相对较好的一个或多个节点，负责本区资源索引、区间资源索引以及区间路由信息的管理与维护。

2　基于位串的分级索引压缩方法

2.1　分级位置索引存在的问题

由LSDS覆盖网络模型可知，系统中任何共享资源(name of resource，NR)都可用一个4级位置标识组成的三元组〈D_{ID},S_{ID},N_{ID},U_{ID}〉唯一定位。其中，D_{ID}为资源所在区标识，全局唯一；S_{ID}为关键字所在站标识，区内唯一；N_{ID}为关键字节点级节点标识，站内唯一；U_{ID}为用户级节点标识。

在用户发起访问后，基于LSDS覆盖网络的完整搜索定位过程为：节点级节点→节点所在站→站所在区→区间→目标区→目标站→目标节点，总体上可分为区间、区内、站内和节点级节点4个范围。区间搜索定位因DAN间的对等关系可采用P2P中广泛使用的DHT(distributed Hash table)方法[12]，共享资源位置索引信息为二元组〈N_R, D_{ID}〉；区内搜索定位则采用分层定位方法，基于分级查询资源位置索引二元组〈N_R, S_{ID}〉、〈N_R, N_{ID}〉和〈N_{ID},U_{ID}〉信息实现。

上述索引和定位方式充分体现了局部自治和分级管理的思想，但是存储空间的浪费是显而易见的。如名为“操作系统”和“数据库”的图书在某站的位置索引信息如表1所示，该表描述了“操作系统”和“数据库”在本站内的节点分布情况。其中，位置是用IP表示的N_{ID}。从该例容易看出，SAN上的资源位置索引记录数为该资源在本站内分布的节点数n_k，在数据表副本冗余度为r_k的情况下，同一资源在上层节点的索引记录总数高达$n_k r_k$。

表1　站内共享资源位置索引信息表

N_R	N_{ID}
操作系统	192.168.1.5
操作系统	192.168.1.6
操作系统	192.168.1.13
操作系统	192.168.1.11
操作系统	192.168.1.3
操作系统	192.168.1.9
数据库	192.168.1.12
数据库	192.168.1.6
数据库	192.168.1.2

2.2　基于位串的分级索引压缩方法

鉴于LSDS中索引网络相对稳定的特点，考虑了利用固定的位串信息来表示资源存储位置的方法。首先，在区级和站级节点上分别建立其子节点集的

一个有序数据结构，然后按如下规则生成一个相应位数的位串：

1) 位串长度为子节点数，每个位对应一个相应顺序的位置(站、节点或用户PC)；

2) 位串初值为全“0”；

3) 资源在某位置上存在则将相应位的值置为“1”，反之则置为“0”。

这样，若干N_R相同的资源位置索引即可压缩为一个二进制位串，反向读取该位串信息即可获知N_R在本区域范围内的位置分布情况。

如在上例中，假设该站有192.168.1.1～192.168.1.14共14个节点，则可相应地生成如图2a所示的14位的位串。假设子节点集构成基于按IP升序排列的单链表，则根据位串与节点间的映射规则，可很容易地判定“操作系统”的目标节点为节点链表的第3、5、6、9、11和13共6个结点对应的服务器节点，“数据库”的目标节点则为节点链表的第2、6和12共3个结点对应的服务器节点。

位 数：	1	2	3	4	5	6	7	8	9	10	11	12	13	14
操作系统：	0	0	1	0	1	1	0	0	1	0	1	0	1	1
数据库：	0	1	0	0	0	1	0	0	0	0	0	1	0	0

a. 补充位前

位 操：	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
作系统数：	0	0	1	0	1	1	0	0	1	0	1	0	1	0	0	0
数据库：	0	1	0	0	0	1	0	0	0	0	0	1	0	0	0	0

b. 补充位后

图2 基于位串的位置索引示意图

利用位串来描述资源的位置索引信息后，共享资源位置索引信息可以得到很好的压缩。如上例中，表1的9条记录可以按该压缩方法压缩为表2的两条记录。

表2 压缩后的索引信息表

N_R	N_{ID}
操作系统	00101100101010
数据库	01000100000100

表3 十六进制表示的位置索引信息表

N_R	N_{ID}
操作系统	2CA8
数据库	4410

为表示方便，进一步按如下步骤将位串转化为十六进制数的表示形式：

1) 确定位串的位数为$\lceil$单位位置总数/8$\rceil\times 8$。其中，单位位置总数在区级节点上是本区内的站总数，在站节点上是本站内的节点总数，在节点级节点上是用户PC数。如上例中站内节点总数为14，则根据上式计算位串的位数为$\lceil 14/8\rceil * 8 = 2 * 8 =$ 16位。

2) 将原位串中位数不足的部分用0补足，如图2b所示。

3) 把二进制串表示为十六进制数。

上例的最终结果如表3所示。

3 共享资源索引信息的发布和汇聚

基于区/站/节点/用户的4级层次关系使相邻两级节点间呈父子关系，父节点通过汇聚子节点发布的位置索引信息实现所辖区域内副本位置索引信息的压缩。

3.1 索引发布

共享资源位置索引信息的发布是一个自下而上的过程，发布过程需要借助本节点上存储的父节点链表信息完成。

3.1.1 父节点链表

为了达到高可靠和高可用等目的，LSDS的资源索引通常发布到多个上级节点。为了明确索引发布节点，需要在子节点中建立一个常驻内存的父节点链表，数据结构如下：

```
Struct Server_Father{
        uint    node_state;//节点状态
        ulong node_ip;//节点IP地址
        server_father *next;
};
```

3.1.2 位置索引发布算法

从单个节点看，索引信息的发布仅从本节点至父节点；从全局来看，索引信息的发布遵循从下级节点到上级节点的原则，完整路线为：(用户级节点→)节点级节点→站级节点→区级节点。

索引发布在节点获取本地及下级资源位置索引信息N_R后，发布内容为N_R和本地节点标识Local_server_ID。算法比较简单，具体如下：

Input: N_R, Server_Father;//共享资源名称、父节点链表

Output: record of (N_R,Local_server_ID)//共享资源的位置索引记录

```
Procedure Index_pulish;//索引发布过程
Traverse_on_Server_Father();//遍历父节点链表
While (server_father.next != NULL)
        publish_to_father   (N_R,Local_server_ID);
Endwhile;
End Procedure;
```

3.2 索引汇聚

索引汇聚主要完成子节点资源索引信息的登记和压缩，需要结合子节点链表顺序进行位串的置位。

3.2.1 子节点链表

子节点链表是上级节点中按序排列的所有子节点地址信息的一种单链表结构，常驻内存，主要用于索引汇聚压缩时按序排位，结构定义类似3.1.1。

3.2.2 索引汇聚算法

索引汇聚主要通过对位串置位的方式实现，置位方法是：遍历子节点链表，获取索引发布节点在位串中对应的相应位并将其置为“1”。如前面示例中关键字“操作系统”的十六进制值为2CA8，对应的二进制为0010 1100 1010 1000。若收到节点192.168.1.6和192.168.1.7发布的该关键字索引信息，则将第6和7位置为1，从而刷新位置索引为2EA8。

汇聚算法如下：

Input: N_R, Publish_server, Server_Subnode //共享资源名称，发布节点，子节点链表

Output: new_index_table //刷新的索引信息表

Procedure Index_Aggregate;//索引汇聚过程

Traverse_on_Server_Subnode (Publish_server);

　　//根据Publish_server遍历子节点链表

If (Publish_server is not found()) then

　　Insert _Server_Subnode (Publish_server);

　　//在子节点链表中按序插入本节点

　　Refresh_index ();//通过移位刷新位置索引

Endif

If (N_R is not found()) then

　　Insert_new_record ();

//插入一条位置索引全0的索引新记录

Endif

Aggregate and refresh index information;

//汇聚并刷新索引信息，置资源对应节点位为1

End Procedure;

4 实　验

本文压缩机制现已应用于面向大规模网络应用的基础软件平台——数字有机体系统[13]。为了验证压缩机制的有效性，基于该系统做了如下两个指标的实验。

4.1 记录压缩比

记录压缩比是用于比较记录压缩结果的一个量化指标。

定义　记录压缩比(record compression ratio，RCR)为共享资源索引信息表中被压缩的记录数和压缩前的总记录数之比。

为了测试压缩机制的RCR，在数字有机体系统某节点上模拟了记录数s分别为10^3、10^4、10^5、10^6和10^7的资源索引信息表的压缩情况，测试用N_R的冗余度r($r\geqslant 1$)在指定范围内随机分布，且平均冗余度如表4所示，压缩结果如图3所示。

表4　测试用资源索引表中的N_R平均冗余度

s	r					
	<3	<4	<5	<6	<7	<8
10^3	1.410 44	1.823 32	2.525 25	2.915 45	3.128 12	3.644 93
10^4	1.420 66	1.873 36	2.522 70	2.788 96	3.008 42	3.752 35
10^5	1.425 56	1.877 62	2.490 28	2.778 86	2.990 55	3.803 29
10^6	1.428 57	1.873 95	2.499 33	2.773 96	3.000 36	3.799 32
10^7	1.428 82	1.875 09	2.499 69	2.777 62	2.999 95	3.801 70

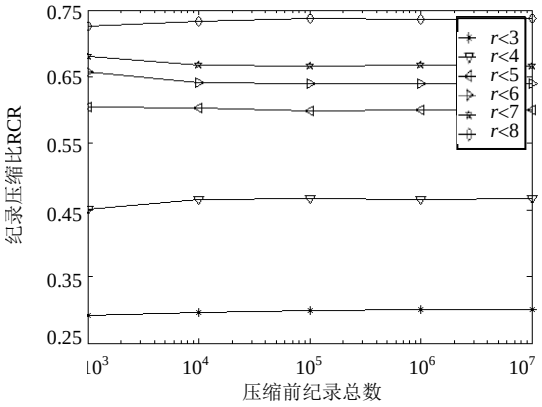


图3　资源索引表的RCR测试结果

测试结果显示，相同冗余度范围下，不同记录数的RCR持平，说明RCR不受记录总数影响；同时，不同冗余度范围下，同一索引表的RCR随平均冗余度的增大明显增大。在本文测试中，当平均冗余度为2.5左右时，RCR可达约60%的高记录压缩比。

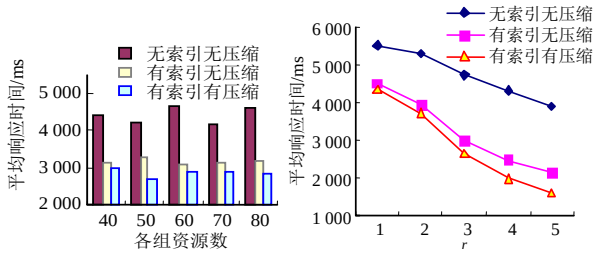
需要说明的是，通过本文压缩机制压缩资源索引记录数后，仍可采用常规的数据表压缩方法进一步降低存储开销。

4.2 副本定位效率

因本文压缩机制作用域为分层的区内，因此，为了验证其对副本定位效率的影响，仅基于数字有机体系统进行了区内副本定位实验，实验环境为：1个含10个SAN的DAN，每个SAN含10个以内RSN，每个RSN的资源索引信息表中含若干本地资源和用户资源索引；测试机器配置为：共15台CPU为P4 2.4 GHz双核、操作系统为Suse 10 Linux(2.6内核)、内存为2 GB的DAN和SAN，其余为P4 2.0 GHz单核、内

存512 KB的PC机；测试资源为：除测试机本地资源外，用户机IP及用户资源通过资源索引信息表虚拟设置，副本冗余度2≤r≤5，资源索引随机分布，最大资源索引数为100 000条；测试方法为：在RSN上调用数字有机体系统提供的CAPI客户机库函数；测试目标为：按共享资源名N_R称获取其节点分布信息。

为避免使用定位结果缓存造成结果失真，在4组不同文件类型的共享资源中各取40、50、60、70和80个不同N_R在无索引、有索引但未压缩和有索引且有压缩3种情况，对平均查询响应时间进行对比实验。同时，取其中一组资源，并配置所有N_R的r分别为2、3、4和5的情况也做相同的实验，结果如图4所示。



a. 4组资源的平均响应时间　　b. 不同r的平均响应时间

图4　副本定位效率比较

实验结果表明，副本定位效率与资源类型无关，与索引、压缩和平均冗余度相关，两种测试情况下，有索引的副本定位效率明显高于无索引的情况，而有索引有压缩的副本定位效率优于仅有索引的情况，并且随平均冗余度增大，优势增强。

5　结语和进一步工作

对采用副本冗余技术的大规模分布式系统来说，副本定位效率往往决定着网络服务性能。针对大规模分布式系统副本定位问题，本文基于物理相邻构建了区/站/节点/用户4级分层的混合覆盖网络，并由相对稳定的区/站/节点3级服务器节点构成索引网络，用于分级管理共享资源索引，在建立上层节点有序子节点集与二进制位串间的映射关系后把相同共享资源的多个位置索引信息汇聚压缩到一个位串，实现了用于副本定位的资源索引分级压缩。在自主研发的大规模分布式基础平台——数字有机体系统的实验和实际应用表明，该压缩机制能够达到很高的记录压缩比，并在一定程度上提高了副本定位效率。

下一步，将结合数字有机体系统所采用的缓存和历史定位记录等辅助机制，进一步合并和压缩各类副本定位信息，以期获得更高的压缩效果和副本定位效率。

参 考 文 献

[1] SAROIU S, GUMMADI K P, GRIBBLE S D. Measuring and analyzing the characteristics of Napster and Gnutella hosts[J]. Multimedia Systems, 2003, 9(2): 170-184.

[2] RIPEANU M, FOSTRE I, LAMNITCHI A. Mapping the gnutella network: properties of large-scale peer-to-peer systems and implications for system design[J]. Distributed, Parallel, and Cluster Computing, 2002, 6(1): 1-12.

[3] 冯劲潇，陈贵海，谢俊元. 分级有序P2P 超级节点拓扑构造[J]. 计算机科学, 2009, 36(10): 127-131.
FENG Jin-xiao, CHEN Gui-hai, XIE Jun-yuan. Hierarchical and ordered P2P super2peer topology construction[J]. Computer Science, 2009, 36(10): 127-131.

[4] 蒋海，李军，李忠诚. 混合内容分发网络及其性能分析模型[J]. 计算机学报, 2009, 32(3): 473-482.
JIANG Hai, LI Jun, LI Zhong-cheng. Hybrid content distribution network and its performance modeling[J]. Chinese Journal of Computers, 2009, 32(3): 473-482.

[5] CHERVENAK A L, CAI M. Applying peer-to-peer techniques to grid replica location services[J]. Journal of Grid Computing. 2006, 4(1): 49-69.

[6] CHERVENAK A, SCHULER R, RIPEANU M, et a1. The globus replica location service: design and experience[J]. IEEE Transactions on Parallel and Distributed Systems, 2009, 20(9): 1260-1272.

[7] BRODER A, MITZENMACHER M. Network Applications of Bloom Filters: A Survey[J]. Internet Mathematics, 2004, 1(4): 485-509.

[8] 陈刚，冯柯，何清法，等. 数据库压缩及解压缩方法. 中国, 39A40B40D, 200410088783[P]. 2007-10-17.
CHEN Gang, FENG Ke, HE Qing-fa, et al. Method of database compression and decompression. China, 39A40B40D, 200410088783[P]. 2007-10-17.

[9] 祝君，林庆农，徐造林. 实时历史数据库中压缩技术的并行化研究[J]. 计算机技术与发展, 2010, 20(7): 36-39.
ZHU Jun, LIN Qing-nong, XU Zao-lin. Research on parallel compression technology in real-time historical database [J]. Computer Technology and Development, 2010, 20(7): 36-39.

[10] 骆吉洲，李建中. 一种有效的关系数据库压缩方法[J]. 软件学报, 2005, 16(2): 205-214.
LUO Ji-zhou, LI Jian-zhong. An efficient compression method of relational database[J]. Journal of Software, 2005, 16(2): 205-214.

[11] LI Zi, MOHAPAIRA P. The impact of topology on overlay routing service[C/OL]//IEEE INFOCOM, 2004[2010-01-15]. http://spirit.cs.ucdavis.edu/ pubs/conf/ infocom04b.pdf

[12] RATNASAMY S, STOICA I, SHENKER S. Routing algorithms for DHTs: Some open questions[J]. Computer Science, Peer-to-Peer Systems Lecture Notes in Computer Science, 2002, 2429(2002): 45-52.

[13] 陈建英，刘心松. 数字有机体数据库系统搜索机制研究[J]. 计算机工程, 2008, 34(4): 45-47
CHEN Jian-ying, LIU Xin-song. Study on search mechanism based on digital organism database system[J]. Computer Engineering, 2008, 34(4): 45-47.

编　辑　漆　蓉