CN43-1258/TP
ISSN 1007-130X

计算机工程与科学
COMPUTER ENGINEERING & SCIENCE

2006 年第 28 卷第 11 期
Vol. 28, No. 11, 2006

文章编号:1007-130X(2006)11-0085-04

一种集群文件系统元数据管理技术*

A Solution to the Metadata Management of Cluster File Systems

李胜利,陈 谦,程 斌,唐 维
LI Sheng-li, CHEN Qian, CHENG Bin, TANG Wei
(华中科技大学计算机科学与技术学院,湖北 武汉 430074)
(School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China)

摘 要:本文研究集群文件系统的特征,提出了一种分布式元数据管理技术。该技术通过哈希方式分布元数据对象、自侦测自适应和连续相邻节点备份的方法,实现了元数据的动态扩展和高可用。在我们研制的 HANDY 文件系统中采用了这项技术。测试结果说明,HANDY 的元数据扩展性是令人满意的,实现了动态可扩展和高可用的设计目标。

Abstract: This paper studies the characteristics on the cluster file systems, and proposes a solution to the management of distributed metadata. The solution has achieved the goals of high availability and dynamic scalability by means of the Self-Detect-Self-Adopt and Neighboring Node Backup. And it had been implemented and tested on the HANDY (High Availability aNd Dynamic scalabilitY) file system, which is designed and developed by ourselves. The test results are satisfactory and reach the design goal of the HANDY cluster file system.

关键词:元数据;并行文件系统;高可用;动态扩展

Key words: metadata; parallel file system; high availability; dynamic scalability

中图分类号:TP311 **文献标识码**:A

1 引言

并行文件系统是构建大规模集群系统的重要组成部分,为应用程序提供高吞吐率的 I/O 以及单一的 I/O 系统映像。目前,并行文件系统往往只追求高性能,忽视了高可用性和动态扩展性。我们设计并实现了一种具有高可用、动态可扩展和高性价比并行文件系统 HANDY(High Availability aNd Dynamic scalabilitY,简称 HANDY)。它所采用的分布式元数据管理技术是本文重点。该技术既弥补了集中式元数据管理的不足,又克服了一般分布式元数据管理策略在可用性上欠缺的不足,为 HANDY 文件系统提供了高可用、动态可扩展的单一元数据系统映像。

2 HANDY 文件系统

2.1 HANDY 系统简介

HANDY 是华中科技大学 CGCL 实验室集群文件系统小组开发的集群环境下的并行文件系统。该系统的设计目标是实现动态扩展性、高可用性、单一系统 I/O 映像和存储路径透明性。HANDY 系统结构如图 1 所示,它采用双环客户/服务器结构。系统提供唯一的根目录作为客户访问入口;文件的具体物理位置对用户完全透明。

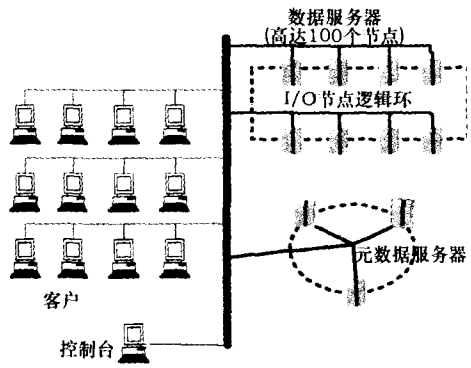

图 1 HANDY 的系统结构图

* 收稿日期:2005-04-18;修订日期:2005-06-21
基金项目:国家 863 计划资助项目(2002AA1Z2102)
作者简介:李胜利(1952-),男,湖北武汉人,教授,研究方向为并行分布式计算、计算机网络、集群与网格计算、计算机软件测试;陈谦,硕士生,研究方向为并行文件系统和集群功能软件。
通讯地址:430074 湖北省武汉市华中科技大学计算机科学与技术学院;Tel:(027)87557047;E-mail:cq_13@sina.com.cn
Address: School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, Hubei 430074, P. R. China

2.2 元数据 handle 概念

元数据是有关数据的数据。HANDY 文件系统以“handle”为元数据对象，handle 是(健，值)二维结构形式的键-值对，标识了一个唯一的元数据。其中，“键”是一个 64 位无符号整数，它由文件的名字和路径哈希得到，起到标识 handle 的作用，而“值”就是元数据的具体内容。系统根据“键”值，将所有 handle 哈希分布到各个元数据服务节点上。

3 元数据高可用性分析

HANDY 文件系统是一个分布式元数据故障可恢复系统，根据生灭过程[1]计算系统可用性。这里定义系统元数据失效，则 $\exists h \in H, h$ 失效。并假设：存储 handle 的节点机发生故障是互不相关事件；在 $(t, t+\triangle t)$ 时间内，任意 handle 及其备份出现故障的概率是 $\lambda\triangle t$；节点恢复概率为 $\mu\triangle t$；故障和修复到达时刻服从指数分布；两个节点在同一时刻故障的概率为 0；每个 handle 在系统里最多有两份拷贝：主副本和备份副本。

假设生灭过程处于状态 n 的稳态概率 P_n 可以由引理给出 $P_n=\frac{\lambda_0\lambda_1\cdots\lambda_{n-1}}{\mu_1\mu_2\cdots\mu_n}P_0$，由此推导出系统的可用度为 $A=1-(\frac{\lambda}{\lambda+\mu})^n$。

显然，一个 handle 的备份节点数越多，handle 的可用性也越高。假设节点平均恢复时间是 0.5 小时，平均无故障时间是 10 小时，由上式计算得到一个 handle，可用度高达 99.999 89% 。

4 元数据管理的技术

4.1 分布式管理的 handle

为了便于管理并保证 handle 数足够使用，“键”值取 $[0, 2^{64}-1]$ 区间上的一个整数。在分布式文件系统中，按照元数据的分布方式，文献[2]提出了基于目录子树结构[3]和完全哈希两种方法。HANDY 采用哈希分布的方法将 handle 映射到各元数据节点。

为适应 handle 分布式管理需要，64 位“键”值的各位赋予了特殊含义。以大端字节序为准，前八位的第一位用于表示元数据的类型，“0”表示数据分片的 handle，“1”表示元数据的 handle；第二至八位是节点的逻辑号，由服务节点网络地址哈希得到；后面的 56bit 信息是 handle 的标识，由文件的路径和文件名哈希获得。

同时，HANDY 系统对 handle 的分配和管理设定了如下三条规则：(1)每个元数据节点负责管理的 handle 范围 MH(Managed Handles，简称 MH)总是逻辑号比它自身小，又比前驱节点逻辑号大的这段范围里所有的 handle；(2)MH 里 handle 的权限包括读、修改 handle 值，但不能分配包含此逻辑号(除了自身逻辑号外)的任何 handle；(3)只有逻辑号是自身逻辑号的 handle，元数据节点才有权分配和删除，该范围的 handle 为元数据服务器可分配 handle，记为 AH(Allocate Handles，简称 AH)，如图 2 所示。

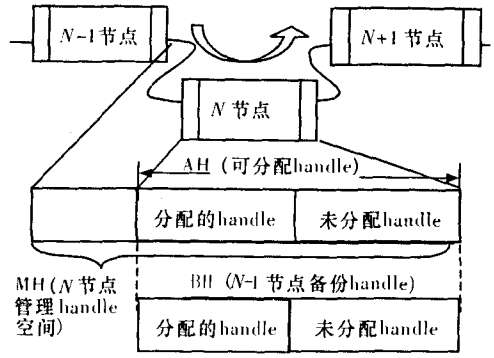

图 2 handle 空间示意图

4.2 元数据高可用的实现

4.2.1 相邻节点备份策略

每个 handle 在系统里有两份拷贝：主副本和备份副本。HANDY 文件系统的备份策略是：任意两个连续相邻节点，后续为前驱作备份，如图 2 所示。

表 1 举例说明了 HANDY 的元数据分配和备份方案。假设每个节点已分配的 handle“键”值范围是 $0 \sim 2^{16}-1$，我们用前四位表示元数据位置等相关信息，一共有四个元数据节点，则元数据管理、分配和备份的情况如表 1 所示。

表 1 HANDY 的元数据分配和备份方案举例

逻辑号	管理 handle 范围(MH)	可分配 handle 范围(AH)	备份的 handle 范围(BH)
#0	0xC0000～0xFFFFF 0x00000～0x0FFFF	0x00000～0x0FFFF	0xB0000～0xBFFFF
#3	0x10000～0x3FFFF	0x30000～0x3FFFF	0x00000～0x0FFFF
#7	0x40000～0x7FFFF	0x70000～0x7FFFF	0x30000～0x3FFFF
#12	0x80000～0xBFFFF	0xB0000～0xBFFFF	0x70000～0x7FFFF

4.2.2 handle 代管及恢复

如果有节点失效，其后续节点开始代管失效节点 AH 范围的 handle。假设有逻辑号分别为 l、m、n($l<m<n$)三个连续元数据节点。当节点 m 失效后，其后续节点 n 在接收到来自 DSP 模块传递来的 m 节点离开的消息后，自动开始代管 m 节点 AH 范围的 handle 。其过程就是将 n 节点 BH 里备份的 H_m' 移动到 n 节点 AH 里，失效的 handle 对系统又变为可见的。接下来，任何有关 m 节点上已分配的 handle 操作全部由 n 节点完成。由于此时不存在逻辑号为 m 的元数据节点，系统不会继续分配包含逻辑号 m 的 handle，但已分配的 handle 仍可以读写。

对系统来说，凡是已分配 handle，必定存在一份拷贝是可用的。备份拷贝在正常情况下不可见，一旦主副本失效，备份就会成为系统可见的。

4.3 元数据动态扩展性的实现

元数据的动态扩展性是通过数据复制和迁移实现的。

4.3.1 元数据节点动态增加

当有新节点加入到元数据逻辑环时，新加入节点会成为加入点处前驱节点的备份节点，这需要完成一次元数据迁移。假设已有元数据节点 m、n($m<n$)，在 T' 时刻有新

节点 NEW 加入到元数据逻辑环上。该节点首先根据网络地址哈希得到一个逻辑号 L_{NEW}，假设 $m<L_{NEW}<n$，则原先的$[L_m+1, L_n]$区间会裂变成$[L_m+1, L_{NEW}]$和$[L_{NEW}, L_n]$两个新区间。也就是说，NEW 节点承担了 n 节点的部分责任，接管了部分 handle 范围。

这时，为了保证新加入节点的前驱节点上备份 handle 位置是正确的，需要将原先在 n 节点上备份的 handle 迁移到 NEW 节点上。迁移工作完成后，NEW 节点成为 m 节点新的备份机；而 n 节点开始备份新节点 NEW 的分配 handle。当这一系列操作完成后，假设为 T''（$T''>T'$）时刻，NEW 节点便向整个系统宣告加入成功。整个过程示意图如图 3 所示。

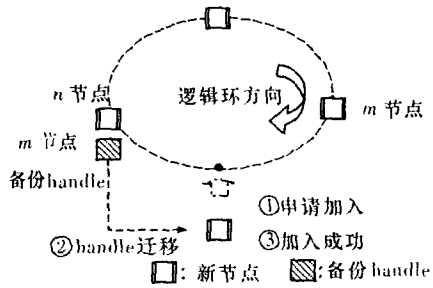

图 3　元数据节点增加示意图

4.3.2　元数据节点动态减少

当有元数据节点动态减少时，由于存在着备份 handle，因此系统的元数据仍然可用。一个元数据服务器的离开，会导致离开点的前后三个节点发生数据迁移。假设有逻辑号分别为 l、m、n、p($l<m<n<p$)的四个连续元数据节点。当节点 m 离开后，节点 n 会接管失效的 handle，并将备份的 handle 升级为系统可见 handle，这一变化前面已经介绍过。这一部分新增的 handle 会在节点 p 上作备份。同时，节点 n 会成为节点 l 新的备份节点。因此，节点的离开总共需要三次数据的复制移动，其过程见图 4。

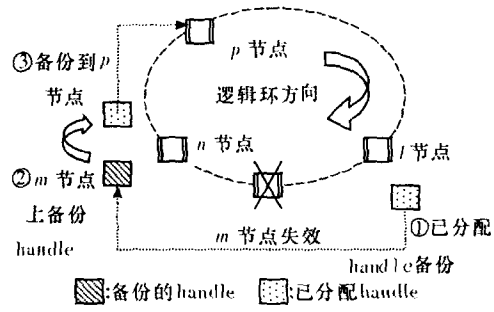

图 4　元数据节点减少示意图

5　测试与性能分析

5.1　测试环境

我们组成由 20 个节点(12 个数据节点，八个元数据节点)构成的 HANDY 文件系统。所有节点硬件配置为 1.6GHz CPU、256M 内存，节点间用 100M bit/Sec 以太网互连。

5.2　测试结果及分析

首先，使用一台客户机测试在根目录下创建、删除文件和目录操作的吞吐率，测试结果如图 5 所示。从图 5 可以发现，HANDY 系统元数据吞吐率峰值过早出现。我们又将测试程序改为在不同目录下执行创建和删除操作，其运行结果如图 6 所示。很明显，元数据操作吞吐率大幅提高，元数据节点扩展性能非常好。

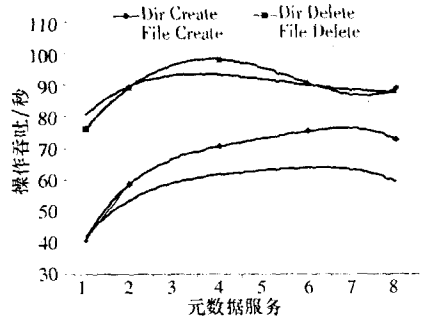

图 5　同一目录下创建和删除的吞吐率

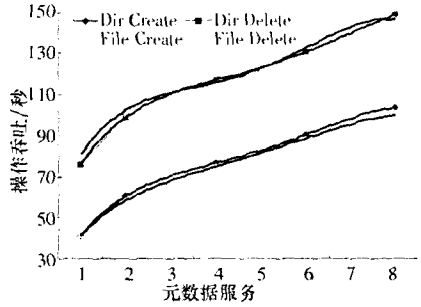

图 6　不同目录下创建和删除的吞吐率

分析这些结果可以得到两点结论：第一，HANDY 文件系统有关元数据操作的性能略优于一般的集群文件系统；第二，HANDY 文件系统在元数据服务器节点数的一定范围内，元数据操作性能会线性增长，实现了动态扩展。

6　结束语

本文提出的分布式元数据管理技术旨在通过逻辑环方式分配元数据对象、自侦测自适应和连续相邻节点备份的方法，实现文件系统中元数据的动态扩展性和高可用性。这一技术已经在 HANDY 文件系统上得到了实现和应用，并且获得了较为理想的测试结果。

但是，整个系统还存在着一些问题。例如，当客户端对元数据的访问集中在某一目录时，必定会造成元数据服务器负载的不平衡，形成系统瓶颈；另外，在系统扩展性方面，元数据吞吐率会随着节点数的增加过早地出现峰值，这与底层网络通讯有关。下一步，我们将研究元数据节点瓶颈所在，进一步提高 HANDY 系统整体性能。

参考文献：

[1] 高文，祝明发. 基于生灭过程的机群系统高可用性分析与设计[J]. 微电子学与计算机，2001，19(4)：47-49.

[2] Scott A Brandt, Ethan L Miller, Darrell D E Long, et al. Efficient Metadata Management in Large Distributed Storage Systems[A]. 20th IEEE/11th NASA Goddard Conf on Mass Storage Systems and Technologies[C]. 2003. 290-298.

[3] Zhihua Fan, Jin Xiong, Jie Ma. A Fialure Recovery Mecha-

nism for Distribted Metadata Servers in DCFS2[A]. High Performance Computing and Grid in Asia Pacific Region[C]. 2004. 20-22.

(上接第 55 页)

易频率对外汇交易行为进行描述,其中对交易金额的离散化是根据外汇局提供的九个金额等级进行划分的。交易频度分两类:不频繁和频繁,交易次数不超过 10 次的为不频繁,交易次数大于 10 次的为频繁。对 1 281×7 的原关系表进行转换之后,形成一个 1 281×33 的具有布尔值的二维表。设最小支持度为 0.01,最小置信度为 0.15,建立的超图模型有 954 个顶点和 9 个超边。超图模型中的顶点数比二值表中的顶点数少,可以看出部分频繁集的支持度小于预先设定的阈值。将超图分割成九类,从聚类结果中可以看出,即期交易总是频繁发生,而远期交易总是不频繁发生的;国内贸易为一般贸易,而国际贸易显出多样性。

7 结束语

本文提出的客户行为分析算法是一种基于超图的聚类算法。它首先根据客户交易行为特点和实际应用价值,对数值属性进行了离散化,并对原始表进行了一种变换,以方便建立超图模型。考虑到外汇交易量很大,要求聚类算法能处理大数据集,同时多维空间中容易出现数据分布稀疏,用多维空间中数据之间的距离尺度来聚类不能产生良好的聚类效果的情况,我们利用超图模型来实现聚类。

参考文献:

[1] 许小满,孙雨耕,杨山等,等. 超图理论及其应用[J]. 电子学报,1994,22(8):65-71.

[2] 魏藜,宫学庆,钱卫宁,等. 高维空间中的离群点发现[J]. 软件学报,2002,13(2):280-290.

[3] George Karypis, Rajat Aggarwal, Vipin Kumar, et al. Multilevel Hypergraph Partitioning: Application in VLSI Domain [a]. The 34th ACM/IEEE Design Automation Conf[C]. 1997. 526-529.

[4] 张蓉,彭奢. 一种基于超图模式的高维空间数据聚类方法[J]. 计算机工程,2002,28(7):54-55.

[5] Eui-Hong(Sam) Han, George Karypis, Vipin Kumar, et al. Clustering in a High-Dimensional Space Using Hypergraph Model[R]. Technical Report TR-97-063, Department of Computer Science, University of Minnesota, 1997.

(上接第 67 页)

部,如果两个高程不相同,取两个高程值的平均值作为这个闭合区的值,如果两个高程相同,就将这个高程值设为此闭合区的值;若不在内部,则表明这个闭合区只由此闭合等值线组成,就将这个高程值设为此闭合区的值,依据原理 1 可知,此闭合区内不会再有其它高程的闭合等值线存在,若有,只可能是属性值相同的闭合等值线。对于最后一条闭合等值线,由于其内部不再有等值线存在,直接用此条等值线的属性值作为闭合区的值。在进行这样的划分后,每个区域都将有一个值,根据这个值来确定颜色,然后用这个颜色填充对应的区域。填充顺序为先填充非闭合区,然后填充闭合区。这样就可以保证闭合区的颜色覆盖非闭合区的颜色,同心的闭合区中,内部的闭合区颜色覆盖外部闭合区的颜色,从而得到不同的区域都能有不同的颜色。在文献[5]中,作者对于闭合区的设置方法为:对于闭合区,因为它只拥有一条等值线,设定这个区域的值为围成它的等值线的高程值[5]。我们认为这种单属性设置方法不太妥当。例如,对于如图 1 所示的一幅等值线图中的四条闭合曲线,若按文献[5]中的方法来填充,高程为 50 的等值线与高程为 70 的等值线之间,在图 1 左半部分将填充高程 50 对应的颜色,而在图 1 右半部分填充高程 70 对应的颜色,从而导致在一幅等值线图中相同的区域赋予了不同的颜色。而按照双属性方法,在高程为 50 的等值线与高程为 70 的等值线之间将填充这两个高程的平均值,左半部分与右半部分颜色填充相同。

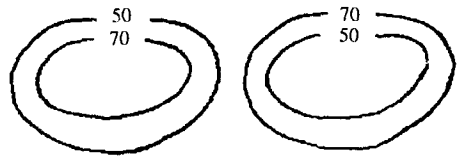

图 1 等值线图中的四条闭合等值线

3 结束语

经过实践证明,本文提出的等值线图区域填充的边界点追踪算法易于实现,便于理解,而且对提出的相关依据给出了证明,从而能够有效地解决等值线图的填充问题。只要再做一些交互处理,就可使用各种图案来填充等值线之间的区域,也可填充指定高程的区域,而且该算法修改后,适用于任意封闭的多边形边界、任何类型的等值线(等高线)的填色。同时,该算法还可应用于气象、地质测绘等领域。

参考文献:

[1] 王伟,袁修孝,张巍. GeoStar 中图形编辑与拓扑关系的建立[J]. 武汉测绘科技大学学报,1995,20(5):36-42.

[2] 李汉林,赵永军. 计算机绘制地质图[M]. 东营:石油大学出版社,1997.

[3] 倪明田,吴良芝. 计算机图形学[M]. 北京:北京大学出版社,1999.

[4] 孙家广. 计算机图形学. 第三版[M]. 北京:清华大学出版社,1998.

[5] 戴常英,李昕,李凌博. 等值线图区域填充的边界扫描算法[J]. 微机发展,2004,14(1):23-25.