CN43-1258/TP
ISSN 1007-130X
计算机工程与科学
COMPUTER ENGINEERING & SCIENCE
2005 年第 27 卷第 8 期
Vol. 27, No. 8, 2005

文章编号:1007-130X(2005)08-00103-03

大规模集群文件系统 LCFS 的元数据管理与访问机制*

The Meta-Data Management and Access Control in the Linux Cluster File System(LCFS)

王召福,章文嵩,刘 仲
WANG Zhao-fu, ZHANG Wen-song, LIU Zhong
(并行与分布处理国家重点实验室,湖南 长沙 410073)
(National Laboratory for Parallel and Distributed Processing, Changsha 410073, China)

摘 要:文件系统的元数据包括文件基本属性信息和目录结构信息。在基于集群技术的大规模文件系统中,有效的元数据管理是系统实现的核心。本文在设计了与元数据管理相关的三类协议的基础上,提出了集群化的元数据服务器实现模型,并分析和比较了各种实现方式的优缺点。

Abstract: The meta-data of file systems include the primary information of files and the directory structure. In the large scale file systems based on the cluster technology, the efficient management of meta-data is critical. Firstly, we define the three classes of meta-data management protocols and propose the clustered meta-data server prototype, and then analyse and compare the merits and demerits of different implementation schemes.

关键词:集群;元数据;文件系统;目录
Key words: cluster; meta-data; file system; directory
中图分类号:TP316
文献标识码:A

1 引言

随着数据规模的增大以及对数据访问性能需求的提高,一种采用元数据与存储设备分离的新型存储体系结构逐渐发展起来,成为存储研究领域的一个重要方向。国外这方面的研究项目如 GFS[1]、Lustre[2]、NASD 等系统,都取得了重要进展。国家 863 课题"大规模可扩展的 Linux 集群文件系统"的目标也正是开展这方面的研究。

集群存储方案的核心是元数据与存储设备的分离,而使用数据的结点在系统中被称为客户端(Client)。随着对象存储设备的迅速发展,后端的存储设备一般采用对象存储系统。采用这个方案实现的系统可以实现 1 000 数量级的客户结点、PB(10^{15})级的存储空间以及 10G/s 级的数据传输速度。我们把这类提供文件服务的系统称为基于对象的大规模集群文件系统。

在基于对象存储的大规模分布文件系统中,一个潜在的系统性能瓶颈是元数据服务器的组织以及客户端对元数据的访问。在这样一个大规模的存储系统中,尽管元数据的数据量相对于整个存储系统的数据容量而言比较小,但有统计表明,在所有文件系统的访问中,对元数据的访问大约占全部访问次数的 50%~80%。所以,高效的元数据管理对整个存储系统提供高性能和可伸缩性至关重要。本文将结合"大规模可扩展的 Linux 集群文件系统"项目的研究,就如何实现大规模集群存储系统中的元数据服务器进行讨论与分析。

2 元数据控制系统的功能设计

2.1 元数据属性模型

传统的文件系统中,元数据与数据本身通常由同一个文件系统管理,保存在同一台存储设备上,并且为提高访问效率,元数据与其描述的数据在物理上尽可能地靠近。而在现有的一些大规模分布式存储系统中,数据可以通过高速网络直接访问。为避免元数据访问成为系统访问瓶颈,

* 收稿日期:2003-09-24;修订日期:2004-06-17
基金项目:国家 863 计划资助项目(2001AA111012)
作者简介:王召福(1975-),男,山东泰安人,博士,研究方向为分布式系统、性能评价等;章文嵩,博士,副教授,研究方向为 Linux 操作系统、集群系统等;刘仲,讲师,研究方向为集群存储系统等。
通讯地址:410073 湖南省长沙市砚瓦池正街 47 号并行与分布处理国家重点实验室;Tel:13560497512;E-mail:wangzhaofu@vip.sina.com
Address: National Labortory for Parallel and Distributed Processing, 47 Yanwachi St, Changsha, Hunan 410073, P. R. China

需要采用多台元数据服务器构成集群来提供系统的元数据服务。

系统中的元数据就是描述对象(文件)的属性,以及描述对象(文件)数据之间关系的信息(目录结构)。属性信息的记录结构主要包括:

(1)扩展文件服务决定:文件的长度、创建时间、读时间、写时间、属性时间(属性的写入或者修改时间)等。

(2)目录服务决定:引用计数(该文件的目录入口数)、所有者、文件类型(目录/文件)、存取权限,集群文件系统的目录结构也在元数据服务器中维护。

2.2 主要协议

在LCFS集群文件系统中,与元数据服务器相关的协议主要有三个:

(1) CFS-MDS:是系统的主要协议,客户端进入文件系统空间首先需要访问元数据服务器。这其中包括资源定位服务、元数据锁机制以及恢复机制等。

(2) OST-MDS:设计为客户服务器协议,其中OST作为客户端,MDS作为服务器,目标是使得OST能够更新元数据服务器需要的属性信息和负载信息。另外,还有体现分布式一致性的恢复协议。

(3) MDS-MDS:各元数据服务器之间的集群特性、数据一致性及分布式锁机制。

与三个协议相关的操作主要包括创建、删除、属性设置及修改等。

2.3 功能模块

因为扩展文件服务的信息是在对象存储系统中维护,因此目录服务与扩展文件服务实际上是分开的。这种分开设计的优点是,因为两个功能关系不密切,保持独立使得系统更加灵活。而带来的问题是,两类"服务器"(元数据服务器与对象存储系统)之间需要更多的通信。

在LCFS系统的元数据服务器上,主要的功能部件如图1所示。

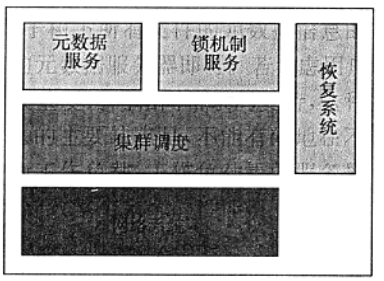

图1 LCFS元数据服务器的组成

各功能部件的主要功能如下:

(1)元数据服务:CFS与MDS之间的主要接口,提供一组用来获取/存放元数据的接口;

(2) 锁机制服务:是系统中最复杂的一部分,用于实现元数据访问之间的同步;

(3)集群调度:实现多个元数据服务器之间的集群化功能;

(4) 恢复系统:出于容错考虑,保证在系统各部分出现故障的情况下恢复到正常工作状态。

在初期的原型系统中,我们主要是实现元数据服务模块部分的功能。

3 集群化的元数据组织

在大规模集群文件系统中,伴随数据量的增加和访问性能要求的提高,必然需要多台服务器来提供元数据服务。目前,研究比较多的是基于集群的元数据服务器系统。

基于集群的元数据服务系统可以从是否共享存储的角度分为两种实现方式:(1)每一个元数据服务器都有自己独立的存储,即整个系统的元数据分为n部分,每一台元数据服务器分管一部分元数据,这可以从存储上减轻单台服务器的系统负载;(2)所有元数据服务器共享存储,即所有元数据服务器共享整个系统的元数据,这主要从系统的处理能力来减轻单台服务器的系统负载。

在第一种方案中,为了实现系统的可伸缩性,需要解决数据的同步、数据的迁移等问题,而如何进行元数据的划分也是需要解决的难题之一。

在第二种方案中,用户根据元数据服务的划分方案选择元数据服务器提供服务。这种方法能够有效地解决不同服务器之间的处理能力负载均衡问题,若进一步采用协作Cache技术,不失为一种有效的集群元数据管理方法。但是,这种方法的缺点是没有在存储一层解决工作负载的平衡问题。这种方案的实现规避了数据同步问题,相对比较容易,而且随着存储技术的发展,存储设备的并行I/O能力也将不断增强。

在现有的分布式文件系统中,在不同元数据服务器之间分配名字空间和工作负载的方法主要有两种,即目录子树分割方法和纯散列方法。

目录子树分割是一种简单、自然、在不同元数据服务器之间分配名字空间的方法,它根据目录子树来分割名字空间,每一个元数据服务器管理整个目录层中的一个或多个子目录树(也称文件集或卷)。NFS[3]、AFS、Coda等都是用这种方法来分割名字空间。目录子树分割方法的一个主要优点是,对该目录子树下所有文件的元数据信息的获取只需要访问该特定的元数据服务器即可。若考虑用户端的缓存机制对用户多次重复访问相同目录的支持,这种方法比较有效。这种方法的主要缺点是,不能有效地在不同元数据服务器之间平衡工作负载,工作负载重的服务器可能成为系统瓶颈,从而导致整个系统的性能下降。通过复制技术在多个服务器上保持副本,能在一定程度上克服这种缺点。但是,副本的增加将导致更多的存储开销,并且维护多个副本的一致性增加了额外的性能开销。

纯散列技术通过散列计算在不同元数据服务器之间分配名字空间,根据文件标示符、文件名或其他相关值的不同散列值来分配文件的元数据到不同元数据服务器。这种方法解决了目录子树分割方法不能有效地在不同元数据服务器之间平衡工作负载的缺点。散列方法不同,性能效果差别很大。散列方法一个明显的优点是,不需要遍历文件路径的所有目录就能直接定位文件的访问权限。问题是,如果使用文件路径名进行散列,则当目录名修改时,该目录下所有的文件的元数据服务器都必须更改,导致大量的元数据迁移。另外,如果不使用目录层次结构,则必须提供一种

机制维护文件的目录层次结构，以提供标准的目录层次语义以及访问文件时确定相应的访问权限。

针对这两种方法的优缺点，我们在 LCFS 中采用混合实现的方案，类似 Lustre[2] 使用文件名进行散列，同时使用遍历目录层次结构来获取文件的元数据。这种策略具有散列方法在不同元数据服务器之间平衡工作负载的优点，但仍有目录层次方法低效的缺点。随着元数据管理元数据服务器的不断扩展和能力增强，我们正进一步针对各种不同系统方案的性能进行建模和定量分析。。

4 结束语

大规模集群文件系统中的元数据管理是海量信息存储的重要研究内容。在 LCFS 系统中，通过设计元数据管理与数据存储的分离，实现对文件系统不同属性的有效管理。随着元数据管理规模的增长，此研究领域还有更多的研究工作。本文分析认为，大规模元数据服务器的组织可以考虑采用集群结构，元数据的划分可以采用基于散列的划分方法。

参考文献：

[1] A Scott, L BrandtEthan, D E MillerDarrell, 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.

[2] Peter J Braam. The Lustre Storage Architecture[M]. Cluster File Systems Inc, 2002.

[3] Brian Pawlowski, Spencer Shepler, Carl Beame, et al. The NFS Version 4 Protocol[EB/OL]. http://www.citi.umich.edu/projects/nfsv4, 2002-05.

[4] Zhou Feng, Jin Chao, Wu Yinghui, et al. TODS: Cluster Object Storage Platform Designed for Scalable Services [EB/OL]. http://www.cs.berkeley.edu/~2f/papers/zf-jc-tods-ica3pp.pdf, 2000-05.

[5] Qin Xin, Ethan L Miller. Reliability Mechanisms for Very Large Storage Systems[A]. 20th IEEE / 11th NASA Goddard Conf on Mass Storage Systems and Technologies[C]. 2003.

(上接第 99 页)

2.4 奇偶文件桶的恢复

发现奇偶文件桶不可用的主文件服务器首先把 Cache 表等查找表的信息发给该奇偶桶；接着把 Pcache 表内的该不可用桶的相应项广播给其它主文件服务器，要求将桶内满足条件的记录插入该奇偶桶；然后，所有服务器根据 Pcache 表、Pminlevel 和 Pmaxlevel 的信息，把桶内满足条件的记录插入该不可用的奇偶桶，即计算每条记录的组关键字 $g(gl,r)$ 经哈希后是否在不可用桶内。

3 性能分析及实验结果

EH * g 比 EH * 需多存储奇偶文件，比 EH * 存储的文件（即主文件）多 $1/k$ 倍。在查询情况下，消息传输的代价与 EH * 相同；偶尔发生不可用时，需要额外的消息。在插入或记录更新时，所需的消息是 EH * 的两倍，因为除要在主文件插入外，还需在奇偶文件进行插入。正常模式下，文件的分裂如同 EH *，无需访问奇偶文件。

实验中，EH * g 文件中主文件与奇偶文件的桶均按 $B+$[4] 树组织，桶的大小均为 50，桶组的大小（即 k）等于 4。初始化时已有四个主文件服务器和一个奇偶文件服务器。关键字 g 通过 $c_g=9*gl+r$ 进行哈希。

实验成功地验证了理论的正确性，即按照上述理论实现系统后，达到了预期的效果。当系统中的一个线程被“当”掉（无论该线程是主文件服务器，还是奇偶文件服务器）时，系统能自动进行恢复，从而实现了高可用性的目的。验证的方法是，当客户机发现服务器不可用时，先放弃该请求，等一定时间（如 2s）后，重新执行该请求，获得了正常的结果。

向文件分别插入 100 和 150 条记录时，只有主文件的 EH * g（即原 EH * 文件）分别需要 30ms 和 40ms 的时间。没有服务器失败的情况下，EH * g 文件分别需要 60 和 90ms 的时间。若发生一次主文件服务器失败并自动恢复的情况下，EH * g 文件分别需要 100ms 和 151ms 的时间。无论插入多少条记录，EH * g 文件都比 EH * 文件的时间高出 25%略多一点。

由此可见，在没有发生服务器失败的情况下，EH * g 文件中插入记录的时间约为 EH * 文件的两倍。在发生一次主文件服务器失败时，则 EH * g 文件中插入记录的时间约为 EH * 文件的三倍多。

4 结束语

在正常的查询代价和适中的存储利用率下，EH * g 为扩展文件提供了高可用性。这对于既需要高可用性又有较多查询的应用而言是很有吸引力的。现代计算机努力把可扩展性、高可用性以及并行性作为其主要特色，EH * g 在这方面应该是很有用处的。

参考文献：

[1] W Litwin, M-A Neimat, D Schneider. LH *: Linear Hashing for Distributed Files[A]. ACM-SIGMOD Int'l Conf on Management of Data[C]. 1993. 327-340.

[2] Victoria Hilford, Farokh B Bastani, Bojan Cukic. EH *: Extendible Hashing in a Distributed Environment[A]. COMPSAC[C], 1997. 217-222.

[3] R Lindberg. A Java Implementation of a Highly Available Scalable and Distributed Data Structure LH * g: [Master's Thesis][D]. University of Linkoping, 1997.

[4] 严蔚敏，吴伟民．数据结构[M]．北京：清华大学出版社，1996.

大规模集群文件系统LCFS的元数据管理与访问机制

作者： 王召福， 章文嵩， 刘仲， WANG Zhao-Fu， ZHANG Wen-song， LIU Zhong
作者单位： 并行与分布处理国家重点实验室,湖南,长沙,410073
刊名： 计算机工程与科学 ISTIC PKU
英文刊名： COMPUTER ENGINEERING AND SCIENCE
年，卷(期)： 2005，27(8)
被引用次数： 3次

参考文献(5条)

1. A Scott. L BrandtEthan. D E MillerDarrell Efficient Metadata Management in Large Distributed Storage Systems 2003

2. Peter J Braam The Lustre Storage ArchitectureCluster File Systems Inc 2002

3. Brian Pawlowski. Spencer Shepler. Carl Beame The NFS Version 4 Protocol 2002

4. Zhou Feng. Jin Chao. Wu Yinghui TODS: Cluster Object Storage Platform Designed for Scalable Services 2000

5. Qin Xin. Ethan L Miller Reliability Mechanisms for Very Large Storage Systems 2003

相似文献(10条)

1. 学位论文 王涌 面向PB级存储系统的元数据集群管理容错方法研究与实现 2007

随着计算机技术，信息技术和互联网络的发展，高性能计算、商业计算、大规模数据处理、信息处理等技术得到广泛的应用。集群系统因其较高的性能价格比和较好的可扩展性而受到越来越多的青睐。与此同时，这些应用对分布式数据存储提出了更大容量，更高性能，更高可用性的要求。

新兴的对象存储结构能够利用现有的处理技术、网络技术和存储组件提供空前的可伸缩性和聚合吞吐量，为构建新一代的大规模并行存储系统提供了基础。

本文在全面深入了解对象存储体系结构与现有对象存储系统的基础上，对基于对象存储体系结构的大规模集群存储系统所涉及的元数据集群管理的容错问题进行了深入的研究，提出了新颖的思想和解决方法。主要的贡献如下：

(1)提出一种面向PB级存储系统的基于目录对象副本的高可用元数据管理模型。通过采用高效的、并发的目录对象副本放置、更新、迁移策略以及管理机制，既保证实现了目录对象数据的可靠性和可用性，又增加了读取数据时的聚合网络带宽，提高读操作的性能。同时采用马尔可夫激励模型进行了定量可用性分析。实际的功能测试与性能测试表明该模型方法能够有效保证存储系统的高可用性，提高元数据服务器集群的整体访问性能。

(2)提出了一种对面向基于日志的元数据管理方法进行检查点操作的思想方法。通过对基于日志的元数据管理方法实施检查点操作，既保证了系统存储空间的有效利用，同时实现了系统的快速恢复，保证了系统的高可用性。实际的测试表明该方法能够充分地利用磁盘空间，提高系统的恢复时间，保证元数据的高可用性，提高元数据的访问效率。

2. 会议论文 刘仲. 王晓东. 周兴铭 可伸缩的元数据集群系统设计 2004

目录子树分割方法与路径名散列方法是目前常见的两种元数据管理方法,它们都存在不均衡工作负载、大量元数据迁移、可伸缩性差等缺点.本文提出一种将元数据划分为目录路径元数据与文件自身元数据分开管理的新的元数据集群系统结构,并设计一种可伸缩的元数据管理模型.

3. 学位论文 秦航 基于集群文件系统的元数据容错研究 2004

为了解决PVFS中元数据管理的瓶颈，高可用性集群文件系统元数据容错系统MDFTS 以PVFS为基础平台，对系统中元数据的故障进行检测与诊断，并进行检查点恢复。为了达到复杂的元数据管理一致性，采用了一个无集中式服务器的体系结构，保证所有的数据和元数据能够存放到系统的任意地方，并且在操作的过程中可以动态迁移；采用元数据的磁盘日志结构和内存日志结构相结合的方式对元数据进行管理，减少了fsck对庞大的文件系统中元数据的扫描时间；为了实现故障恢复，提出了元数据容错的设置检查点算法和回卷恢复的算法，提高了文件系统元数据服务的可用性；给出了基于元数据故障的随机过程模型，可以通过减少检错时间提高文件系统的可用度。系统在操作系统应用层实现，通过修改元数据结构和相关的系统调用，使得集群文件系统内部各个数据节点和元数据管理节点相互协作，统一调度，支持高可用性。
测试结果表明，元数据容错系统可以针对系统模拟的不同类型的故障进行错误检测，并能够对系统和应用进行切换与恢复。

4. 期刊论文 王召福. 章文嵩. 刘仲. WANG Zhao-fu. ZHANG Wen-song. LIU Zhong LCFS中元数据服务器的可靠性分析模型 -计算机工程与科学2005,27(5)

可靠性问题是研究大规模集群存储系统的一个重要方面,元数据服务器是大规模集群存储的中心.本文针对基于镜像复制和共享存储的服务器实现方案,采用马尔可夫补偿模型研究元数据服务器的状态迁移概率,分析了元数据服务器集群的可靠性以及数据一致性对可靠性的影响,对实现大规模集群文件系统中的元数据服务器有重要的指导意义.

5. 学位论文 于洪芬 小规模集群文件系统的元数据管理策略研究 2007

集群文件系统是集群的一个重要组成部分，它为用户提供一个虚拟化大容量存储器的统一访问接口和高I/O带宽。由于集群文件系统的文件数据分散存储在各个节点上，文件的定位需要借助元数据来完成，虽然元数据的数据量相对于整个存储系统的数据容量而言比较小，但有统计表明，在所有文件系统的访问中，对元数据的访问大约占全部访问次数的50%～80%，因而元数据的管理成为了管理数据的一个关键。
本文主要研究小规模集群文件系统的元数据管理策略，给出了一种具体实现元数据控制系统的方案，即两级元数据服务器结构，它由高级元数据服务器（Advanced Metadata Server，AMDS）和双元数据服务器（Double Metadata Server，DMDS）构成，分别管理目录元数据和文件属性元数据，实现数据的层次管理。在访问元数据时，分两级访问：第一级是访问AMDS中目录路径元数据，在文件系统名空间中定位元数据；第二级是访问DMDS，对文件自身元数据进行操作。在元数据控制系统中，AMDS是主体，所有的元数据操作请求都是通过AMDS处理的，只在需要时访问DMDS。这样可以保证元数据服务的高可靠和高扩展，同时能保证元数据访问能够一次定位，提高系统性能。具有容错能力的双元数据服务器采用双系统备份、镜像复制存储技术，从而保证了元数据管理的高可靠。

6. 学位论文 何飞跃 并行文件系统元数据管理研究 2004

随着高性能微处理器、高速网络的出现和对计算能力需求的增大,以廉价硬件和软件支撑的集群系统越来越被广泛地使用,引起集群技术的迅猛发展.集群文件系统是集群的一个重要组成部分,作为一种集群体系结构上的并行文件系统,它为用户提供一个虚拟化大容量存储器的统一访问接口和高I/O带宽.由于集群文件系统的文件数据分散存储在各个结点上,文件的定位需要借助元数据来完成,元数据的管理成为管理数据的一个关键.为了提高元数据管理的可靠性,需要具有容错能力的元数据管理系统.为此,我们针对集群文件系统的元数据管理,设计了一个双元服务器系统.该系统内部由两台元数据服务器组成,通过对元数据的镜像产生副本,保证元数据的可靠性;通过主服务器失效后从服务器接管服务来屏蔽故障,保证元数据服务的连续性.系统具有集中管理方式控制简单、易于实现和维护等优点,克服了其单一失效点的缺陷,同时又避免了分布式管理的一致性维护设计与开销.在Linux内核空间实现元数据镜像技术、故障检测技术、IP接管技术和恢复技术,具有对应用程序透明性的特点.系统的最终目的是将其结构推广到多机情况下,进一步提高容错能力,实现高可用性.为了提高元数据服务器的处理效率,提出了一种寄生式元数据存储管理方法.并行文件系统的元数据寄生在本地文件系统内核中,通过增加系统调用实现对寄生元数据的操作,保证对现有系统的兼容性.将该方法应

用于PVFS(Parallel Virtual File System)的元数据管理,元数据操作性能提高大约5～8倍.

7.期刊论文 田俊峰.于洪芬.宋玮玮.TIAN Jun-feng.YU Hong-fen.SONG Wei-wei 小规模集群文件系统中两级元数据服务器的设计与实现 -小型微型计算机系统2007,28(6)

在集群文件系统中,元数据服务器是整个系统正常运转的核心,它的可靠性和性能是设计系统时需要着重考虑的问题之一.本文设计了一个具有高可靠性、高性能的两级元数据服务器系统,兼顾了集中式元数据管理和分布式元数据管理的优点.系统中高级元数据服务器负责维护文件系统全局的目录结构和管理整个文件系统的命名空间,双元数据服务器负责维护文件元数据的分布信息,并采用了马尔可夫回报模型对两级元数据服务器系统进行了可靠性分析.实验数据表明,具有两级元数据服务器的集群文件系统能提供高吞吐量.

8.学位论文 黄九鸣 SAN环境下高性能集群文件系统研究与实现 2006

本文在对当今主流的网络存储技术进行研究的基础上，针对SAN环境提出了一种集群文件系统模型SANFS，并对该模型进行了设计实现和测试验证。

SANFS具有高性能、高可扩展能力、低成本、易架设等特点，适用于非线性视频编辑、科学计算、VOD视频服务等以顺序I/O为主且对数据I/O稳定性要求较高的领域。

SANFS模型基于CIFS协议构建，可被实施于当今主流的各种操作系统。模型充分利用SAN网络上各主机共享存储设备的优势，让各客户端在元数据服务器的控制下，直接对存储设备进行I/O操作，使系统I/O吞吐率和I/O速率稳定性达到一个较高水平。模型以元数据“分散缓存集中控制”和文件数据“分布读写”为主要指导思想，其核心技术包括：块设备对象缓冲、文件映射关系缓冲、元数据预分配及预取、SANFS OpLock机制及SANFS DOOR Lock等。其中，经本文抽象定义的SANFS DOOR Lock模型可被应用于其它场合的缓冲器中，具有较大参考价值。此外，SANFS还在客户端的文件系统一级，实现了对元数据服务器虚拟卷的支持，提高了系统可扩展性。

本文还详细设计了SANFS客户端在WINDOWS上的实现与元数据服务器在Linux上的实现，并研究与分析了其中的关键技术。

最后，通过对SANFS当前实现版本的测试，验证了模型的正确性和各种技术手段的有效性。

9.期刊论文 李胜利.陈谦.程斌.唐维.LI Sheng-li.CHEN Qian.CHENG Bin.TANG Wei 一种集群文件系统元数据管理技术 -计算机工程与科学2006,28(11)

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.

10.学位论文 邵强 对象存储文件系统中元数据管理集群关键技术研究与实现 2005

在信息时代，数据存储具有举足轻重的地位，存储已经开始成为关系企业生存发展的重要因素，如何构建一个高性能、高可伸缩、高可用、易管理、安全的存储系统成为目前所面临的一个重要课题。

基于对象的存储技术是存储领域的新兴技术，提出了一种新型的存储结构。对象是这种存储结构的核心，封装的元数据和文件数据分别由不同的系统管理。元数据包括文件的属性和访问权限，由元数据服务器管理；文件数据条块化存储于智能的对象存储设备；客户文件系统向用户提供存储系统的接口，可以与元数据管理系统交互和与对象存储设备直接进行数据交换。基于对象存储结构构建的大型分布式文件系统，可扩展性强、性能高，可提供较强的并发数据处理能力。

本课题主要研究对象存储结构中的元数据管理。元数据服务的扩展性和高性能对于对象存储结构至关重要，采用集群管理元数据是大型存储系统中元数据管理的一种趋势。本文采用一种新颖的结构实现层次管理元数据的元数据管理集群，分别以目录路径索引服务器集群和元数据服务器集群管理目录元数据和文件元数据，并研究其中的关键技术。

在研究集群负载均衡的基础上，设计和实现元数据管理集群静态负载分配与动态反馈重分配相结合的负载均衡方案。通过静态元数据分割算法，实现元数据服务负载分流或者元数据分布存储实现负载分流；服务器动态反馈服务器负载信息，实现不均衡负载重新分配。这样保证元数据管理集群的负载均衡，并解决“热点”数据访问问题。

另外，研究元数据管理集群中可用性问题，DPIS集群中采用共享容错磁盘阵列和节点容错机制解决共享存储数据和节点故障问题，MDS集群采用备份服务器保证服务器节点出现故障时元数据服务工作的接替和数据备份的重建，实现元数据管理集群在单点失效和特定的多点失效情况下的容错和恢复，保证系统的可靠性和可用性。

引证文献(3条)

1.刘群.冯丹 基于层次结构的元数据动态管理方法的研究[期刊论文]-计算机研究与发展 2009(z2)

2.田俊峰.于洪芬.宋玮玮 小规模集群文件系统中两级元数据服务器的设计与实现[期刊论文]-小型微型计算机系统 2007(6)

3.田俊峰.于洪芬.宋玮玮 小规模集群文件系统中两级元数据服务器的设计与实现[期刊论文]-小型微型计算机系统 2007(6)