CEPH动态元数据管理方法分析与改进

冯幼乐[1] 朱六璋[2]
(1. 中国科技大学信息科学技术学院自动化系 2. 安徽电力继远软件公司)

摘 要：分布式文件系统(CEPH)的动态元数据管理方法极大地提高了元数据服务器的性能和扩展性。本文首先分析了CEPH元数据服务器集群中的负载均衡策略，针对其在异构元数据服务器和网络延迟较大时存在的问题提出了改进方案。实验证明，改进后的方法不仅提高了系统的性能，扩大了系统的使用范围。

关键词：分布式文件系统；元数据管理；负载均衡

Analysis and Improvement of CEPH Dynamic Metadata Management

Feng Youle[1] Zhu Liuzhang[2]
(1.Dept. of Automation, University of Science and Technology of China 2.Anhui Electric Power Jiyuan Software Co. Ltd)

Abstract: The dynamic metadata management of CEPH has significantly increased the performance and scalability of metadata server cluster. We first introduce the migration algorithm of directory subtrees, and then propose an improved algorithm to resolve the problem existed in larger heterogeneous MDS and network latency. With the improved migration algorithm, not only the migration process is less called and unnecessary network cost is saved, but also the performance in large latency network is improved, which makes CEPH can be used in more environments.

Key words: distributed file system; metadata management; load balance

0 引言

分布式文件系统(CEPH)通过将多台机器的资源组织起来，对外提供统一的、大容量、高性能、高可靠的文件服务，满足了大规模应用的要求，是目前存储领域研究的重点和难点。CEPH[1-3]通常由元数据服务器(MetaData Server,MDS)集群和存储服务器集群构成。统计表明：在文件系统的访问中，对元数据的访问次数占全部访问次数的50-80%[4]。为应对大量的元数据操作请求，保障良好的性能和扩展性，元数据的管理方式与MDS集群的负载均衡策略极其重要。

现有的元数据的管理方式主要为集中存储分布式处理[2,5]和分布式存储分布式处理[1]两种方式。第一种方式元数据保存在共享的存储设备中，元数据和MDS的对应关系是动态划分的，每台MDS负责缓存一部分目录子树并处理相应的元数据操作；动态划分在出现热点数据或者MDS负载过高时可以很方便地进行目录子树的复制和迁移，易于扩展MDS和负载均衡，但实现较为复杂。第二种方式元数据按一定的方式分布到MDS上，每台MDS负责处理存储在其上的元数据请求，元数据与MDS的对应关系一旦确立就不会改变；这种方式实现较为简单，元数据的划分一般采用静态子树划分或者hash方法。由于目录子树并不是均衡增长，静态子树划分很容易出现负载不均衡；hash方法在初始的时候，可以通过良好的设计使得元数据在MDS间均匀分布，整体负载比较均衡，但是在增加或减少MDS的时候，需要重新调整hash函数，这两种方法在负载均衡时都会导致大量的数据迁移。

CEPH[5]采用集中存储分布式处理的动态元数据管理方法，通过对目录子树的复制和迁移实现了MDS的负载均衡，具有良好的性能和扩展性。本文首先分析了CEPH元数据服务器集群中的负载均衡策略，针对其在异构元数据服务器和网络延迟较大时存在的问题提出了改进方案。实验证明，改进后的方法提高了系统的性能，而且扩大了系统的使用范围。本文组织如下：第1节介绍CEPH文件系统的系统架构及其元数据管理器集群的负载均衡策略；第2节针对其在异构和高网络延迟环境下存在的问题提出改进方案，第3节对其进行实验分析，最后给出总结。

1 CEPH文件系统架构及其元数据服务器集群负载均衡策略

1.1 CEPH文件系统体系架构

CEPH采用元数据和文件数据分开处理的体系结构，由三个子系统组成：客户端(CLIENT)，元数据服务器集群(Metadata Cluster)和对象存储集群(Object Storage Cluster)。MDS维护全局的名字空间，负责处理元数据相关的请求以及相应的权限管理；对象存储设备负责文件数据和元数据的存储，为客户端和MDS提供统一的数据读写服务。在CEPH中，元数据保存在对象存储设备中，MDS利用缓存的数据对外提供服务。由于MDS本身并不存储数据，所以可以很方便地进行目录子树的复制迁移以实现负载均衡。

图1 CEPH文件系统架构[6]

1.2 元数据的负载均衡策略

1.2.1 基本定义

节点负载(1)：采用一段时间内的CPU占用率的平均值作为节点负载的度量。

元数据热度(p)：每当有来自用户的元数据请求时，对应的元数据热度增一。考虑到不同时间内的元数据请求对热度的影响应该是不同的，元数据的热度会随时间衰减。元数据访问的时间间隔越长，对热度的影响应该越小。对某一元数据访问时，其在目录树上的祖先节点的热度亦会受到影响，即热度的更新会向上传播，传播的量逐层衰减。综合以上考虑，得到元数据节点的热度

计算公式如下：

$$p_{new} = p_{old} * f(\Delta t)+1$$

$$p_{ancestor\_new} = p_{ancestor\_old} * f(\Delta t)+1/2^n$$

其中Δt为当前时间与上次计算热度的时间差，f为衰减函数，n为被访问元数据与祖先节点的目录层次差。

1.2.2 子树复制

当元数据的热度超出阈值后，将会启动子树复制流程，系统在其他的MDS上创建缓存副本。子树复制可以解决热点数据的访问问题

1.2.3 子树迁移

MDS定期上报自己的负载信息，当单台MDS的负载持续一段时间高于平均负载一定范围时，将会启动子树迁移流程：

(1)根据收集到的全局负载信息，选择一负载较轻的MDS作为子树迁移的目标MDS，根据过载MDS和轻载MDS的负载与平均负载的差值决定迁移负载的数量；

(2)根据过载MDS的整体热度与负载的关系，决定迁移目录子树的热度；

(3)从过载MDS中选择相应热度的目录子树进行迁移。

子树迁移策略解决了负载在MDS间分布不均的问题。

2 CEPH元数据管理方法的改进

CEPH的元数据管理策略可以很好地应对热点数据访问以及负载分布不均的问题，但其实现的子树迁移算法却有如下问题：

(1)迁移算法默认了各MDS能力相同。实际上同样的元数据访问在不同MDS上造成的负载是不同的。目录子树从高负载、能力强的MDS上迁移到低负载、能力弱的机器上时，能力弱的MDS可能接受过多的元数据负载，很快成为新的系统瓶颈；虽然在新一轮的迁移活动中，目录子树会从能力弱的MDS迁移出去，但仍然浪费了大量的网络流量，延长了负载均衡的时间。当目录子树从高负载、能力弱的MDS上迁移到低负载、能力强的机器上时，虽然不会造成网络流量的浪费，但实际上可以迁移更多的热度过去，没有充分利用迁移的机会。

(2)子树迁移算法的目标是达到全局负载的平均，在网络延迟较大的情况下，这一目标的实现需要花费较大代价。由于网络延迟的影响，收集全局负载信息可能需要较长的时间，而且在选择迁移目标时，并没有考虑网络延迟的状况，最后选出的目标对象间可能传输代价较大，极大地影响了负载均衡算法的效率。

本文将针对以上两个问题对子树迁移算法进行改进，在迁移时综合机器能力与网络状况合理选择迁移目标和热度，提高了迁移性能，扩大了CEPH的使用范围。

2.1 基本定义

节点负载的重新定义(L)：节点负载除了与CPU有关外，还和内存占用、带宽和I/O等有关。考虑到CEPH元数据服务器的特性，我们采用CPU，内存和带宽占用情况的加权和重新定义负载。其中l_{cpu}、l_{mem}，l_{bw}分别为CPU、内存和带宽的占用情况，α、β、γ为相应的加权系数，根据应用类型的不同通过反复试验得出。

$$L=\alpha*l_{cpu}+\beta*l_{mem}+\gamma*l_{bw}$$

节点能力的评估(s)：节点能力一般用其处理器能力、内存容量等参数的加权和来表示，但这种方法需要针对每台MDS做实验才能得出具体的加权系数，不同应用系数还有改变。本文采用MDS的元数据总热度(P)与负载(L)的比值来定义节点能力。容易看出，处理的元数据请求越多，产生的负载越小，节点能力越大，这是与实际状况相符的。

$$s=P/L$$

传输代价(c)：MDS节点间传输单位数据需要花费的代价。大部分节点间c的计算在两台服务器有数据交换的时候捎带完成，考虑到网络结构的稳定性，c的更新并不频繁。c还可以通过手动配置直接指定，对于新加入的MDS，通过指定其到某些MDS的传输代价，可以加快其融入MDS集群的速度。

MDS的分区：通过对c进行聚类，将MDS分成不同的区域，区域内MDS间的传输代价明显小于同区域外MDS通信的传输代价。区域内距离聚类中心最近的MDS作为决策MDS，负责收集局域内MDS的信息和迁移决策。

2.2 子树迁移

MDS定期向区域内的决策MDS上报自身的负载和热度，当单台MDS的负载持续一段时间高于区域内平均负载一定范围时，将会启动子树迁移流程：

(1)根据过载MDS负载L_i与区域平均负载$\bar{L}$的差值计算需要迁移出的负载L_{out}。

(2)根据负载和热度信息，计算第j台轻载MDS能接受的最大负载ΔL_j，$\Delta L_j=(\bar{L}-L_j)\frac{P_i}{L_i}\frac{L_j}{P_j}$，其中$P_i$、$P_j$分别为过载MDS和轻载MDS的热度信息。

(3)选择能接受负载最多的MDS作为目标MDS，迁移的负载量为相应的ΔL_j与L_{out}中较小者。

(4)通知过载MDS迁移目标MDS和迁移的负载，过载MDS中选择相应热度的目录子树进行迁移。

3 实验及结果分析

在实验中，元数据管理系统有4台异构的MDS组成，其中MDS0和MDS1性能相同，MDS2和MDS3性能相同，MDS0的性能是MDS2的十倍。分别在两种网络环境下进行实验：首先4台MDS连接在同一个局域网内，比较两种负载均衡方法的性能；接下来MDS0和MDS2在同一局域网，MDS1和MDS3在同一局域网，两个局域网通过Internet连接起来，比较两种负载均衡方法的性能。通过向元数据管理系统连续发送6个小时的数据，记录负载均衡过程中各MDS的实时负载，迁移次数与通讯量等数据进行比较。

表1 算法性能比较

网络环境	迁移算法	迁移次数	迁移通讯总量	迁移花费时间	操作延时
局域网	改进前	12	50M	15min	小
	改进后	10	40M	12mn	小
Internet 连接的局域网	改进前	12	50M	40mn	大部分操作延时较大
	改进后	22	35M	15mn	小部分操作延迟较大

在表1可以看出，在局域网环境下，改进后的算法起到减少迁移次数，降低网络通讯的作用。在广域网环境下，原有的算法在广域环境下迁移元数据，迁移流程耗时较长，而且元数据的跨区域分布，使得许多元数据操作需要跨区执行，延迟较大。改进后的算法使得元数据仅在区域内迁移，仅有少部分操作涉及区域外的数据，大部分操作在区域内即可完成，提高了用户体验。在广

域环境下，迁移次数较多是因为两个区域内同时进行了数据迁移。

4 总结

本文对CEPH元数据的负载均衡算法进行了研究，并提出一种改进方法。改进的算法通过对元数据服务器进行区域划分，在子树迁移时综合考虑负载和机器性能，优化了子树迁移目标的选择策略。实验证明，改进后的方法在异构元数据服务器和网络延迟较大的情况下，仍能发挥较好的性能，扩大了CEPH的适用范围。

参考文献:

[1] Braam P J.A Scalable,High-Performance File System [M]. Lustre Whitepaper Version 1.0,2002.
[2] 黄华. 蓝鲸分布式文件系统的资源管理[D]. 博士学位论文，中国科学院计算技术研究所, 2005. 5.
[3] Ghemawat S,Gobioff H,Shun-Tak L.The Google file system[C]//Proceedings of the Nineteenth ACM Symposium on Operating Systems Principles,2003 Oct 19-22,New York. New York:ACM Press,2003:29-43.
[4] Roselli D,Lorch J,Anderson T.A comparison of file system workloads[C]//Proceedings of the 2000 USENIX Annual Technical Conference,San Diego,CA,June 2000. USENIX Association:41-54.
[5] Sage A.Weil.Ceph.Reliable,scalable,and high-performance distributed storage[D].Santa Cruz:University of California, December,2007.

作者简介:

冯幼乐，男，1986年生，硕士研究生，研究方向为网络多媒体
手机：13645512517
电子信箱：fengyl03@163.com
联系地址：安徽省合肥市中国科学技术大学西区7号楼621（230027）
朱六璋，男，1969年生，安徽舒城人，高级工程师，从事电力应用软件研发技术管理、企业信息化项目相关技术应用研究工作

基金项目:

国家863课题"新一代业务运行管控协同支撑环境的开发(2008AA01A317)"资助

(上接第8页)

在考虑时间因素调整权重后，最相似案例为X2007X15。为简单起见，对得到的最相似案例使用空调整。调整前得经济损失为18540万元，与实际损失误差50.73%；调整后得经济损失为14340万元，与实际损失误差16.59%。时序调整后的权重在计算相似度上考虑较全面，得到的结果较未考虑时间因素的更接近实际值。

3 结论

本文针对案例推理灾害救助系统提出了一种基于时序的权重调整算法，考虑了自然灾害各特征属性对时间的敏感性，在寻找相似案例时，检索出的案例更为合理。但该算法本质上没有摆脱k-NN方法运行时间复杂度较高的缺点，随着案例库中案例数目的增加，算法复杂度将成指数增长。这时应该考虑对案例库建立归纳索引，这将在以后的工作中进行探讨。

参考文献:

[1] 杨 健, 杨晓光, 等. 一种基于k-NN的案例相似度权重调整算法[J].计算机工程与应用, 2007(23):12-15.
[2] 武民民, 宋良图. 基于替代算法的案例推理灾害救助系统[J]. 计算机系统应用, 2009, 18(4):13-15.
[3] 常春光, 崔建江, 等. 案例推理中案例调整技术的研究[J]. 系统仿真学报, 2004(6):149-154, 172.
[4] 蹇 明, 黄定轩. 无决策属性的多属性决策权重融合方法[J]. 西南交通大学学报, 2005(2):134-138.

作者简介:

姜枫(1984－)，男(汉族)，江苏常州市人，中国科学院物质科学研究院合肥智能机械研究所在读硕士研究生，研究方向为模式识别与智能系统。
电话：13637098541
电子信箱:raksasa@live.cn或jfjf@mail.ustc.edu.cn
通信地址：安徽省合肥市1130信箱8楼（230031）

基金项目:

1. 国家科技支撑计划：中国重大自然灾害风险防范技术，项目编号2008BAK50B08.
2. 国家科技支撑计划：灾害应急决策支持与远程会商协同技术研究，项目编号2008BAK49B05.

CEPH动态元数据管理方法分析与改进

作者： 冯幼乐， 朱六璋
作者单位： 冯幼乐(中国科技大学信息科学技术学院自动化系)， 朱六璋(安徽电力继远软件公司)
刊名： 电子技术
英文刊名： ELECTRONIC TECHNOLOGY
年，卷(期)： 2010，47(9)
被引用次数： 0次

参考文献(5条)

1. Braam P J.A Scalable,High-Performance File System[M].Lustre Whitepaper Version 1.0,2002.

2. 黄华.蓝鲸分布式文件系统的资源管理[D].博士学位论文,中国科学院计算技术研究所,2005.5.

3. Ghemawat S,Gobioff H,Shun-Tak L.The Google file system[C]//Proceedings of the Nineteenth ACM Symposium on Operating Systems Principles,2003 Oct 19-22,New York.New York:ACM Press,2003:29-43.

4. Roselli D,Lorch J,Anderson T.A comparison of file system workloads[C]//Proceedings of the 2000 USENIX Annual Technical Conference,San Diego,CA,June 2000.USENIX Association:41-54.

5. Sage A.weil.Ceph.Reliable,scalable,and high-performance distributed storage[D].Santa Cruz:University of California,December,2007.

相似文献(10条)

1. 学位论文 冯幼乐 分布式文件系统元数据管理技术研究与实现关键技术研究的功能衰退 2010

分布式文件系统通过将多台机器的资源组织起来，对外提供统一的、大容量、高性能、高可靠，易扩展的文件服务，满足了大规模应用的要求，是目前存储领域研究的重点和难点。针对文件系统中元数据和文件数据存储和访问的不同特点，分布式文件系统通常包括数据存储系统和元数据管理系统。为应对大量的元数据操作请求，保障元数据操作的一致性和可靠性，元数据管理系统的设计与实现及其重要。本文设计并实现了分布式文件系统LandFile的元数据管理系统，研究了分布式文件系统元数据管理的关键问题以及解决策略，具体内容如下：

研究了面向节能的元数据服务器集群的负载均衡策略：大规模的应用中，能源问题越来越成为人们关注的焦点。本文在动态元数据管理策略的基础上，提出了一种面向节能的负载均衡策略，通过在负载较低时，主动关闭元数据服务器集中负载，达到整体节能的目的。随着负载的增加，再逐步打开元数据服务器对外提供服务。良好设计的动态元数据管理策略保证了服务器加入和退出时的平滑扩展。实验表明面向节能的负载均衡策略在负载低谷时，节能效果非常明显，对系统性能亦影响较小。

研究了元数据管理的一致性保障策略：分布式条件下元数据可能存在多个副本，许多元数据请求可能涉及多台元数据服务器，本文设计了基于主节点的元数据缓存架构，实现了多副本元数据的更新策略，针对分布式操作，设计并实现了基于两阶段提交的协议来保证其一致性。

研究了元数据管理的可靠性保障策略：本文研究了元数据服务器的节点管理、故障检测以及故障恢复机制，实现了区域自治的节点管理方法和基于日志的可靠性保障策略，保证了元数据服务器单点失效时，能够被快速替换和恢复，失效时系统仍能无间断的提供服务。

利用上述研究的负载均衡策略，一致性保障策略与可靠性保障策略，设计并实现了元数据管理系统，包括元数据请求处理和可靠性模块，改进了负载均衡模块。

所研究实现的元数据管理子系统，是863课题“新一代业务运行管控协同支撑环境的开发（2008AA01A317）”中分布式文件系统LandFile的重要组成部分。

2. 学位论文 华清 网络环境下分布式文件系统的设计与实现 2006

今时今日，网格技术已经不再把自己局限在高性能计算的范畴中，而是通过向网络服务靠拢，建立起一套面向服务的体系架构。相应于此，网格中的数据模块也不再是一个提供资源的底层支持模块，而逐渐向一个功能独立的，相对自治的分布式文件系统发展。

本文的目的在于构造一种网格环境下的分布式文件系统。在结构上它相当于中国教育科研网格底层支撑平台的数据模块部分，为CGSP其他功能模块提供持久的数据存储功能以及稳定而高效的数据传输服务。另一方面，这个系统也可以独立运行，网格终端用户可以通过它建立自己的数据空间。

传统的分布式文件系统往往是紧耦合的、基于文件级别应用副本策略的。这些系统虽然性能出众、但往往是应对专门系统设计，通用性不强。而过去基于Web的分布式文件系统往往性能低下、存储能力弱、传输效率低。为了满足网格环境下海量数据密集型应用的需求以及面向服务的框架，我们设计并实现了一个基于分片的、松耦合的分布式文件系统，它包括底层存储资源集合、存储资源管理模块、元数据管理模块、副本管理模块、数据传输管理模块、信息监控模块、策略分析模块以及虚拟文件视图终端等部件。

本文设计并实现的系统具有以下的特点和优势：

2 稳定性：通过副本管理模块调整系统中文件合理的冗余度，对”零副本危机”进行预测并予以避免。策略分析模块会在存储资源选择时挑选健壮性最佳的数据节点进行存储；

2 高效性：在传输中使用GridFTP带状并行传输，同时提高服务器端和客户端的带宽利用率。策略分析模块会在构造传输计划时挑选最近、当前可用带宽最大的节点进行传输；

2 通用性：通过使用网络服务资源框架实现远程调用，存储资源只需向存储资源管理模块汇报就可以加入资源集，用户也可以在任何终端登入自己的用户空间；2 可扩展性：通过对策略分析模块的合理设计，开发评估函数接口。

他模块可以通过调用接口得到所需要的结果，而管理员可以通过对接口的不同实现，对配置参数的调整改变系统的运行策略。这样系统也便于重构和功能扩展。

本文所有设计、实现的结果都在一个网格环境下测试，并通过OptorSim模拟了大规模节点数、长运行时间的运行环境进行可用性测试。

3. 学位论文 王建勇 可扩展的单一映象文件系统 1998

传统的分布式文件系统不能为机系统提供严格的单一映象功能,而且由于不能适应计算技术的发展趋势,无法满足应用对机群系统的I/O性能、可扩展性和可用性的需求.曙光超级服务器是典型的机群系统,他们为其研制开发了可扩展的单一映象文件系统COSMOS,并称其原型系统为S2FS.该文主要描述了S2FS的设计、实现及评价.首先,S2FS是一个全局文件系统,它通过实现位置透明性和严格的UNIX文件共享语义而保证了严格的单一系统映象.其次,为了提高S2FS系统的性能和可扩展性,该文对合作式缓存进行了研究和评价.最后,为了避免单一服务器瓶颈问题,我们为S2FS采用数据存储与元数据管理分开的

策略,实现了分布式的数据存储和元数据管理功能.虽然该文在保证系统单一映象和二进制兼容性的基础上,对适合于机群文件系统的可扩展性技术进行了研究,但由于应用对I/O的需求是永无止境的,且其I/O存取特征以及计算技术的发趋势也有不断发生变化,这一切都为我们未来研制新型的分布式文件系统提出了更大的挑战.

4.学位论文 李健灿 集群存储系统架构及相关技术的研究与实现 2009

随着计算机技术的不断发展和应用，人们对数据存储的需求越来越大，对存储的容量和速度的要求越来越高。传统的存储系统因其物理组成而受到很大的限制，集群存储作为一项已被广泛使用的技术，能够提供按比例增加的服务器或存储资源的性能、容量、可靠性及可用性，突破了单机设备的种种限制。

在存储技术飞速发展的今天，集群存储技术还有着许多急需解决的问题，如随着存储的迅速增长，存储的可扩展性问题，还有存储的安全性，存储的备份和恢复等问题都还没有得到很好的解决方案。

本文研究了当前存储体系结构及相关技术，对网络连接存储、存储区域网络和集群存储文件系统的架构、特点和相应的解决方案作了较详细的分析，在此基础上，开发了protoDFS集群分布式文件系统，初步实现了文件存储、文件同步、文件访问等功能和大容量存储的特性。

本文重点讨论了集群存储的元数据分布查找方式和数据副本分发方法。元数据是最重要的系统数据，元数据的访问性能影响着并行文件系统的性能。如何合理的将系统中的元数据布局到不同的MDS上，从而使得对元数据的请求相对均匀地分散到各个MDS上是元数据分布的目标，本文尝试提出一种提高系统可扩展性和负载均衡性的解决方案。另外，数据副本的分发和访问也是系统容错性、可靠性的重要内容。本文研究了动态副本管理的技术和副本一致性策略并进行了实现。

5.会议论文 贾瑞勇.张延园 SAN文件系统元数据服务器集群体系结构及关键技术研究

SAN文件系统是一种基于存储区域网络的分布式文件系统,其设计目标是保持传统分布式文件系统的文件共享语义,同时可达到接近于本地文件系统的性能.元数据服务器集群是提高SAN文件系统性能、可扩展性和可用性的关键.本文给出了一种新颖的元数据服务器集群模型,该模型具有良好的可扩展性和容错能力,并能提供高度并发的元数据服务;深入探讨了实现该模型的两大关键技术:组通信和元数据管理,最后简单介绍了目前国际上的发展现状和研究趋势.

6.学位论文 郭威 分布式文件系统ZD-DFS的设计与实现 2006

随着互联网迅速发展，对互联网海量数据的存储和读取成为诸多网络应用系统的重要负载。当文件个数，读取需求急剧增加时容易导致后台文件系统服务器负载过大而成为整体性能的瓶颈。而传统的文件系统很难满足海量数据存储和读取的性能要求，而现有的通用分布式文件系统并不专为互联网应用背景的海量小文件存储提供良好的支持。所以需要开发对互联网环境下的海量小文件支持良好的分布式文件系统。本文设计实现的分布式文件系统ZD-DFS采用以元数据服务器管理后台存储服务器的形式向应用屏蔽资源定位、负载均衡、分布式事务、数据迁移、数据一致性等细节，并提供一个统一的编程接口。该系统通过元数据管理服务器和存储服务器节点提供满足Web应用需求的功能。元数据管理服务器负责管理分布式文件系统的全局信息，实施资源定位，数据迁移和负载均衡，管理维护后台存储节点，提供对分布式存储、分布式事务、文件资源读取等功能的全局支持。在元数据服务器的管理协调下，提供了对客户接口机对各种文件的存储，读取，更新，删除操作。ZD-DFS的分布式事务处理支持两阶段提交协议，实现了XA协议，对事务处理过程中的故障和常提供了容错机制。ZD-DFS同时通过锁机制，版本戳和心跳协议等方法保证了分布式的数据一致性。ZD-DFS分布式文件系统为海量数据提供了良好的存储、读取、更新性能，系统各部分性能均衡，不存在明显的性能瓶颈。具有较好的可扩展性，能够方便地进行存储和计算能力的扩展，为大型网络应用提供了较好的底层支持。

7.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。

8.学位论文 娄成龙 对象文件系统中元数据管理技术研究 2005

随着信息技术的进一步发展，以及网络的大规模应用，带来了数据的爆炸性增长，也给网络存储带来了巨大的发展机会。如何构建一个扩展性强、可靠性高、易管理的高性能存储系统成为目前研究的一个重要课题。

基于对象的存储技术是存储领域的新兴技术，它提出了一种新型的存储结构，数据对象是这种存储结构的核心，数据对象封装了用户数据(文件数据)和这些数据的属性(元数据)，他们分别由不同的系统管理。以对象存储结构为基础构建的大型分布式文件系统，可扩展性强、可靠性高，能提供较强的并发数据处理能力。元数据服务管理在对象存储文件系统中尤为重要，采用集群管理元数据是大型对象存储系统中的一种趋势，本文致力于研究对象存储结构中的元数据集群管理技术，所做的主要工作如下：1.分析研究基于对象存储系统的体系结构，设计并实现了一个小型的对象存储文件系统原型OCFS。

2.研究对象存储文件系统中的元数据管理，设计原型改进的文件系统OCFSⅡ，对元数据管理集群实行层次化管理，分别以目录路径索引服务器DPIS集群和元数据服务器MDS集群管理目录元数据和文件元数据。

3.在研究集群负载均衡的基础上，设计和实现OCFSⅡ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和元数据分布存储，实现元数据服务负载分流；采用动态反馈服务器负载信息，实现不均衡负载重新分配。保证元数据管理集群的负载均衡，并解决了“热点”数据访问问题。

4.设计实现了OCFSⅡ元数据管理集群可用性保障方案。目录路径索引服务器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；元数据服务器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。实现了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，保证了系统的可靠性和可用性。

9.学位论文 侯玮玮 基于内容存储关键技术研究与实现 2007

随着信息化程度的不断提高，数据对于企业的重要性凸现，存储技术在其中起到的作用日益增加，而网络技术的发展以及数据量的飞速增长，需要新的存储网络技术适应现有的网络和存储环境。面向对象存储网络这种新的存储网络体系结构，结合SAN和NAS的优点，逐渐成为学术界和工业界关注的热点。基于内容存储是面向对象存储的一个典型范例，除了具有面向对象存储的优点之外，还具有数据压缩和保持数据完整性等优点。

本文研究了基于内容存储的关键技术，包括对象和对象ID生成方法、元数据管理、存储空间管理、数据压缩技术和WORM技术。在此基础上完成了基于内容存储网络CASN的总体设计，并提出了新的安全模型和协议。同时，本文还设计并实现了一种基于内容的存储设备CASD。CASD使用根据对象内容得到的Hash值作为存储对象的标识，并以此来访问存储对象。这种方法具有一次写的功能，防止意外或者恶意的数据破坏。此外，CASD还能去除重复数据，可以节约存储空间和节省网络带宽。以上这些特性使得CASD具有很广泛的应用前景。本文介绍了基于内容存储的主要应用场景，并提出了一种基于内容存储技术在分布式文件系统中的应用。

本文在CASD的基础上，搭建了基于内容存储网络原型系统。为了进行对比测试，搭建了Intel OSDN、NFS和iSCSI三种网络存储系统，并分别比较了CASN系统和这几个存储系统的性能。实验结果表明，CASN系统和其他网络存储系统相比，性能具有一定的优势。

10.学位论文 刘安 PC机群上的分布式协作化文件系统DCFS的设计与实现 2000

由于当前的各种机群分布式文件系统中缺乏一种适合于PC机群环境的、面向高性能的实现,研究人员为DISCOS设计和实现了可扩展的单一映象文件系统-DCFS.该文描述了DCFS的设计、实现和性能分析,着重讨论了设计中的一些关键性问题.DCFS首先是一个单一映象的分布式文件系统,支持数据的透明存取和共享访文件共享语义.另外,研究从员在修改Linux操作系统核心的前提下, 通过实现一个新的基于SIO规范的I/O库,保证了与其底层平台的无缝连接,以及和Linux应用程序的完全二进制兼容.DCFS还是一个可扩展的分布式系统,在设计上采用无集中服务器Serverless结构, 将数据存储和元数据管理工作分布到整个系统中,并实现了分组的stripe存储,提高了系统的带宽和扩展性.DCFS的设计主要是面对并行应用,通过对可扩展I/O的研究,研究人员在DISCOS机群上实现了可扩展I/O底层编程界面(SIO-LLAPI)的一个子集.