分布式存储系统中元数据系统的研究与设计

苏　勇[1,2]　周敬利[1]　姜明华[1,2]　刘　钢[1]
[1](华中科技大学计算机科学与技术学院,武汉 430074)
[2](武汉科技学院计算机科学学院,武汉 430073)

摘　要　在分布式存储系统中,元数据服务系统是一个潜在的访问瓶颈。文章提出了一种基于分布式哈希函数和共享存储器思想的元数据服务器系统,并且与 Lazy Hybrid(LH)方法进行了对比研究,通过仿真测试表明其具有较高的元数据操作吞吐量和减少了元数据服务器之间元数据移动及易扩展等特性。

关键词　元数据　分布式存储系统　Lazy Hybrid　对象存储

文章编号 1002-8331-(2006)20-0105-03　**文献标识码** A　**中图分类号** TP393

The Research and Design of MDS in Distributed Storage System

Su Yong[1,2]　Zhou Jingli[1]　Jiang Minghua[1,2]　Liu Gang[1]
[1](Department of Computer Science, Huazhong University of Science and Technology, Wuhan 430074)
[2](Department of Computer Science, Wuhan University of Science and Engineering, Wuhan 430073)

Abstract: Metadata server system is a latent visit bottleneck in distributed storage system.This paper introduces a new metadata server system based on distributed hash and shared storage, and compare to Lazy Hybrid.By Simulation test, it indicates that it has characteristic of a better metadata operation throughout, reduces metadata shift between the metadata server, easy expansibility and so on.

Keywords: metadata, distributed storage system, LH, Object-Based Storage

1　引言

在大型分布式存储系统中为了获取高性能和高扩展性,避免瓶颈是非常重要的。虽然相对于整个系统的存储量,元数据的存储量是非常小的,但是元数据的访问量占整个系统的访问量的 50%~80%[1],因此对元数据的访问是一个潜在的瓶颈。

在现在比较流行的对象存储系统中,分离了数据和元数据的管理,不像传统的文件存储系统,其数据和元数据都由同一台机器存储和管理。对象存储设备(Object-based Storage Device)管理底层的存储,例如对象到块的映射和请求调度,提供一个对象访问接口,取代了以前的块级访问接口。其元数据服务器(Meta Data Server,MDS)是一个单独的机群,它管理着系统的名字空间、目录层次和文件及目录的许可,并且实现文件到对象的映射。MDS 的管理是否高效率将决定存储系统的整体性能[2]。

在 MDS 设计中一个关键的问题是如何把系统中的元数据分配到每个元数据服务器上,并且具有高性能和高扩展性。目前有两种比较流行的元数据分配技术:一个是目录子树分配方法[3]。它根据目录子树,把系统的目录树元数据分配到每个 MDS 上。这种方法的最大的缺点是当某个文件或目录成为访问热点时,将使访问某个 MDS 成为严重的瓶颈。另一种方法是使用哈希(Hash)函数,把元数据的名字空间分配到 MDS 机群中[4]。Hash 函数根据文件标示符、文件名或者其它相关值,把元数据分配到不同的元数据服务器上,但是也必须保持目录层次结构以获得标准的目录层次语义及各层次目录和文件的访问权限。Hash 函数方法比目录子数分配方法具有较好的负载均衡。其中 Lazy Hybrid (LH) 方法可算是当前比较先进的元数据管理方法,它也采用 Hash 函数根据完整路径参数来分配元数据到不同的 MDS 服务器上,避免大量访问单个 MDS 服务器的热点产生,另外当目录和文件名以及访问许可发生该变时采用 lazy 更新策略,达到更有效的元数据的管理。

本文设计了一种新的元数据 Hash 管理方法(Metadata Hash Partition,MHP),它也采用 Hash 函数的方法来分配元数据到 MDS 机群中,相比 LH 方法,采用了两次 Hash 方式进行元数据分布,元数据存储在共享的网络存储器上,在仿真测试中其元数据操作的吞吐量略优于 LH,并且其最大特色在于减少了在 MDS 机群中 MDS 之间的元数据的移动及良好的扩展性。

2　Lazy Hybrid(LH)

LH 使用 Hash 函数的方法在元数据服务器机群中分配存储元数据[5],其元数据访问原理如图 1 所示。

其中 MLT 为元数据查询表(Metadata Look-Up Table,MLT),它是一个容量较小并且较少更新的全局表,保存着 Hash 映射值与 MDS ID(MDS 机群中每个 MDS 用 ID 编号)的映射,如表 1。客户端通过文件名的 Hash 值查询 MLT,获得需访问的元数据所在的 MDS ID,然后向查询到的目的 MDS 发送元数据请求,最后由这个 MDS 对客户端的请求作出响应。

基金项目:国家自然科学基金资助项目(编号:60373088);国家重点实验室项目(编号:51484040504JW0518)

作者简介:苏勇,博士生,研究方向为计算机网络存储。周敬利,教授,博导,研究方向为计算机多媒体网络通信,高性能网络接口和计算机网络存储。姜明华,博士生,研究方向为计算机网络存储。刘钢,博士生,研究方向为计算机网络存储。

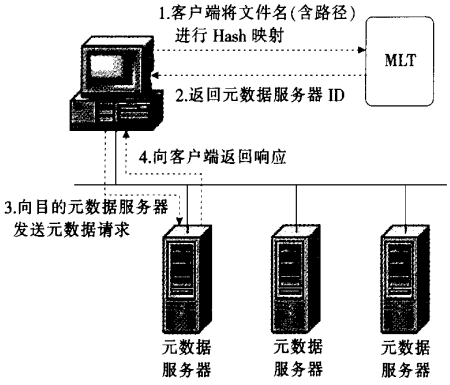

图 1 元数据访问机制

表 1 元数据查询表示例

Hash 值范围	元数据服务器 ID
0-2FFE	0
2FFF-5FFE	1
5FFF-8FFE	2
8FFF-BFFE	3

LH 的优点:采用 Hash 方法,比直接用目录子树分配方法能更好地进行负载均衡,避免了某个目录或文件成为热点后,造成某单个 MDS 成为访问瓶颈;采用 lazy 更新策略,当某个目录或文件的元数据发生改变时,LH 并不是立即对其相关的所有的元数据进行更新,而是只发出无效消息,当无效的元数据再被访问时才进行更新,这样将元数据修改时产生的大量同步更新操作,分散到以后若干时间内完成,减少了瓶颈产生的机率。

LH 的最大缺点是当 MDS 机群中 MDS 的加入或移出时,将会造成大量的元数据在 MDS 之间的移动,这对 MDS 服务乃至存储系统性能会有很大的影响。

3 MHP 系统设计

MHP 也使用 Hash 方法在 MDS 机群中分配元数据,另外最为关键的是采用所有 MDS 共享一个公共的存储池的方法。这个公共存储池划分成若干个逻辑分区,这些逻辑分区再分配给 MDS 机群中每个 MDS 管理,建立逻辑分区到 MDS 多对一的映射。MHP 系统可分为两层模型,由高到低分别是映射管理(Mapping Manager,MM)和逻辑分区管理(Logical Partition Manager,LPM),如图 2 所示。

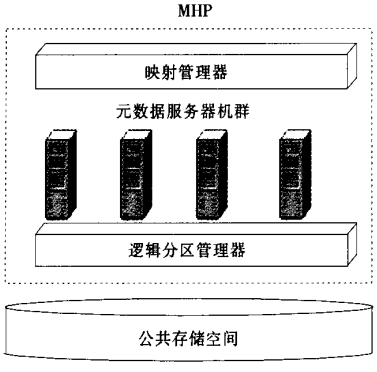

图 2 MHP 模型图

在 MHP 系统中元数据访问过程如下:首先,MM 模块将文件名通过 Hash 函数得到一整数值,这个值将映射到一个保存有文件的元数据的逻辑分区号;接着,映射管理器根据这个逻辑分区号映射到唯一的 MDS,然后客户端将直接向该 MDS 发送含有文件路径 Hash 值的查询请求;最后 MDS 访问公共存储池中的逻辑分区,将请求的元数据返回给客户端。

3.1 映射管理

客户端要想在 MDS 机群中获得文件的元数据,必须要知道两方面的信息:(1)哪个 MDS 管理元数据;(2)元数据在逻辑分区中的定位。映射管理器通过两种 Hash 求值得到这两种信息:(1)对文件名的 Hash 求值,然后映射到逻辑分区号,再查询 MLT 可获得 MDS ID,从而知道是哪个 MDS 管理元数据;(2)然后将路径 Hash 求值可用来在 MDS 上定位所需的元数据。比如客户要访问一个文件"/a/b/fi lec",那么客户首先对"filec"进行 Hash 求值来判断在 MDS 机群中由哪个 MDS 管理其元数据,然后再对"/a/b/filec"进行 Hash 求值用来在 MDS 上定位其元数据。另外对于某些常用的文件名,如"help"、"setup"等,其 Hash 值必须进行处理,要使其尽量分布到不同的 MDS 上,这样可避免不同目录下的常用文件名的文件高频率的访问造成的瓶颈。MHP 的处理方法是对常用文件名的文件在选取 MDS 时,不是用文件名进行 Hash 映射而是用路径进行 Hash 映射,这样可避免不同目录下常用文件名的文件元数据集中在某台 MDS 造成瓶颈。

映射管理器的工作原理可用下列公式(1)和(2)描述:

$$Pi=f(H(filename)) \quad Pi\in\{1,Pn\} \tag{1}$$

H 表示对文件名的 Hash 函数,f 表示对 Hash 值进行逻辑分区映射,Pi 表示逻辑分区号,Pn 表示逻辑分区总数。

$$MDSi=mlt(Pi) \quad MDSi\in\{1,MDSn\} \tag{2}$$

MLT 表示对逻辑分区 Pi 映射到某台元数据服务器 $MDSi$,其功能主要是通过查询 MLT 实现,MLT 表结构见表 2。

表 2 元数据查询表示例

逻辑分区号(Pi)	元数据服务器(ID)
0~10	0
11~20	1
21~30	2
31~40	3

在 MHP 中,客户端通过对文件名的 Hash 求值映射到目的 MDS ID,然后只向一个目的 MDS 发出元数据的查询请求,这符合目前 MDS 机群中普遍采用"a single message to a single metadata server[6]"的原则。

3.2 逻辑分区管理器

逻辑分区管理器负责管理一个独立的公共存储空间,这个公共存储空间可由网络存储器构建,如 NAS(Network Attached Storage)等设备,并且将它划分成若干个逻辑分区。每个逻辑分区存储全局元数据中的一部分,并且唯一的分配给一个 MDS 授权访问。MDS 通过逻辑分区管理接口只能访问其托管的逻辑分区。

4 MHP 的负载均衡设计

在 MDS 机群中,客户端对元数据的请求访问必须要考虑负载均衡的问题。在 MHP 系统中负载均衡涉及两个方面:(1)MDS 的负载均衡;(2)逻辑分区的负载均衡。由于元数据是通

过文件名 Hash 来选择了逻辑分区存储，而一个 MDS 要负责若干个对逻辑分区元数据访问请求管理，因此 MDS 形成瓶颈的概率要比逻辑分区大得多。基于这个原因，在 MHP 设计中我们主要考虑 MDS 的动态负载均衡。MDS 的动态负载均衡通过以下公式(3)、(4)描述：

$$MPWi=\sum_{i=1}^{Mn} PWi/Mn \tag{3}$$

$$PW(i+1)=PW(i)*\alpha+PWcurrent(1-\alpha) \tag{4}$$

其中 $MPWi$ 表示 ID 为 i 的 MDS 的负载；Mn 表示 MDS 的总数；PWi 表示逻辑分区 i 的负载，负载以其被访问的频率作为参数；$PW(i)$ 表示在 i 时刻时逻辑分区的负载，$PWcurrent$ 表示在当前时刻的逻辑分区的负载；α 为 0 到 1 之间的固定参数，α 用来平衡逻辑分区负载的旧值和当前值对负载新值的影响。当 α 趋与 0 时，逻辑分区负载主要取决与当前负载，当 α 趋与 1 时，逻辑分区负载的新旧值变化很小，不能及时反映当前负载值，参考网络通信中 RTT 计算采用的经典 α 值，我们这里也取 α 为 $\frac{7}{8}$。

在动态负载均衡实现中，PWcurrent 由映射管理器每隔 60s 周期进行计算，然后通过公式(4)计算 PWi，再用公式(3)计算各个 MDS 的负载。当每个 MDS 负载相当或者 MDS 负载没有超过设置的极限值时，则不需要进行负载均衡，否则则要对 MDS 托管的逻辑分区进行重新分配，分配原则按照公式(1)进行，然后刷新 MLT。MLT 由 MDS 机群中一台特别的 MDS 维护，并且及时的将每次更新的 MLT 分发给客户端。

5 MHP 与 LH 对比

MHP 与 LH 的共同点是，两者都采用 Hash 函数进行元数据分发，但是在 Hash 方式、故障恢复和扩展性方面 MHP 比 LH 有较大的优势。

5.1 Hash 方式

LH 方法在 Hash 时，采用直接对文件的完整路径进行 Hash 求值，通过这个 hash 值来选取 MDS 进行元数据分配或元数据查询。而 MHP 采用两种 hash，先只对文件名进行 hash 求值，用来选取 MDS，然后与选取的 MDS 再用完整路径名的 hash 进行元数据的分配或查询。两者相比，MHP 能更好地将同一路径下的文件的元数据分配到不同的 MDS 上，这样可有效避免当某一目录成为访问热点时造成某一 MDS 成为瓶颈。

5.2 故障恢复和扩展性

LH 方式一般采用元数据直接存储在 MDS 上，这样一旦某台 MDS 发生故障或 MDS 机群负载超过阀值需要扩展时，必然需要将原 MDS 的元数据重新分配到新增加的 MDS 上，这是一种 MDS 之间元数据的物理迁移，这对存储系统的整体性能会有较大的影响。

而 MHP 采用的是由网络存储器组成的公共存储空间，当某台 MDS 发生故障时，其原先托管的逻辑分区将由映射管理器重新进行映射，映射按照公式(3)进行，由其它的 MDS 接管它的逻辑分区，这样 MDS 机群不会因为某台 MDS 的故障而不能正常工作。并且这种故障恢复是不需要人工干预，而且也没有 MDS 之间元数据的物理迁移的。

在扩展性方面会遇到两种情况：(1)MDS 机群处理元数据请求的负载超过某一阀值时；(2)元数据存储超过公共存储空间的阀值。以上两种情况都需要对 MDS 机群进行扩展，但是扩展的方式不同。在第一种情况下 MHP 可以通过增加 MDS 来解决，增加 MDS 后仍然只需要映射管理器按照公式(3)进行重新映射，分担元数据请求负载，而且也不需要 MDS 之间元数据的物理迁移。但是在第二种情况下，就需要增加网络存储器对 MHP 的扩展，而且这种扩展需要对新增加的逻辑分区进行元数据的物理迁移，为了避免这种情况需要在构建 MHP 时做好规划，在成本允许的前提下尽量保证有较大的公共存储空间。

6 仿真测试

为了比较 MHP 与 LH 在提供元数据访问时的性能，本文用 NS2 进行了模拟仿真。仿真测试的重点是在元数据服务器提供元数据操作的吞吐量性能测试。在 NS2 模拟仿真中，本文用开源项目的 DiskSim[7]工具对系统的本地存储器和网络存储器进行仿真，网络采用千兆以太网的高速网络的性能指标。仿真测试中采用 trace - driven 的方式，使用的 trace 是基于 2003 年 Lawrence Livermore National Labs 集群文件系统的 trace[8]。该 trace 构建了大量客户端结点访问同一文件或者同一目录下的文件的爆发性访问。测试结果如图 3。

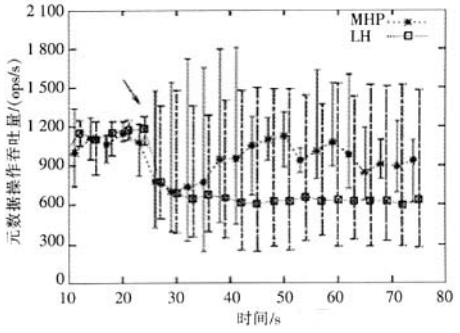

图 3 元数据操作吞吐量

图 3 所示为在动态的工作负载下，一段时间内 MDS 的吞吐量的变化，可见在相同的负载情况下 MHP 的吞吐量性能要比 LH 要好。

7 结束语

本文针对分布式对象存储系统，在基于分布式哈希方式下提出了一种新型的基于分布式哈希方法和共享存储器思想的元数据管理方法 MHP 系统。相比 LH，在仿真测试中其元数据操作吞吐量略优于 LH，并且由于采用共享存储器结构使其具有良好的 MDS 扩展性，减少了在 MDS 机群中 MDS 之间的元数据的移动。今后，准备构建 MHP 的原型系统，在实际的工作环境中对其性能进行测试，并作进一步完善，使其提供高效的元数据服务。(收稿日期：2003 年 12 月)

参考文献

1.D Roselli，J Lorch，T Anderson.A comparison of file system workloads[C].In：Proceedings of the 2000 USENIX Annual Technical Conference，2000：41~54

2.P J Braam.The Lustre storage architecture.2002

3.E Levy，A Silberschatz.Distributed file systems：Concepts and examples[J].ACM Computing Surveys，1990；22(4)

4.P F Corbett，D G Feitelso.The Vesta parallel file system[J].ACM Tran-

(下转 115 页)

(3)职员接收到所有的 T_i 后验证其有效性如下：

$$g^{T_i?}=g^{k_iR}\left(g^{L_i(\sigma_i+\omega_i')}\right)^{h(m,R)}\ \bmod p=$$

$$r_i^R\left(\left(y_0^{h(K,PGID)}\widetilde{K}^n\prod_{j=1}^{t-1}A_j^{ID_i^j}\right)^{L_i}W^{L_i}y_i\right)^{h(m,R)}\ \bmod p$$

若上式成立，职员计算 $T=\sum_{i=1}^{t}T_i \bmod q$。$\{m,K,R,T,PGID\}$ 即为消息 m 的签名。

4.4 代理签名验证阶段

首先计算：

$$T=\sum_{i=1}^{t}T_i=\left(\sum_{i=1}^{t}k_i\right)R+\left(\sum_{i=1}^{t}x_i+\sum_{i=1}^{t}L_i\sigma\right)h(m,R)=$$

$$\left(\sum_{i=1}^{t}k_i\right)R+\left(f(0)+f'(0)+\sum_{i=1}^{t}x_i\right)h(m,R)=$$

$$\left(\sum_{i=1}^{t}k_i\right)R+\left(\sum_{j=1}^{n}(\tilde{k}+\alpha_i)+\sum_{j=0}^{n}x_ih(K,PGID)+\sum_{i=1}^{t}x_i\right)h(m,R)$$

则 $g^T=R^R\left(K\left(y_0\prod_{i=1}^{n}y_i\right)^{h(K,PGID)}\prod_{i=1}^{n}y_i\right)^{h(m,R)}\ \bmod p$ 成立。

5 安全性分析

改进后方案的安全性建立在单向函数和离散对数问题是难解的这一安全性假设之上，下面我们讨论几个可能的攻击：

(1)我们来看一下合谋攻击：假如 t 个代理签名人合谋实施类似于 Sun 方案的合谋攻击，他们应用拉格朗日插值多项式容易计算出 $f(0)$ 和 $f'(0)$，也就是能够分别重构原始签名人分发给每个代理签名人的秘密份额 σ_i 和代理签名人执行可验证的秘密共享方案后所分享的代理份额 ω_i'（也就是得到 σ）。尽管如此，在个人签名产生式 $T_i=k_iR+(L_i\sigma+x_i)h(m,R)\bmod q$ 中仍然包含两个未知参数 x_i 和 k_i，而且 x_i 和 k_i 的求解都建立在求解离散对数问题的难解性假设之上，所以他们无法得到参与代理签名人的个人签名，也就是说合谋攻击无法实现。

(2)再来看一下伪造攻击：假设恶意的攻击人想伪造一个任意选取的消息 m' 的有效代理签名。令 $\widetilde{Y}=\left(y_0\prod_{i=1}^{n}y_i\right)^{h(K,PGID)}$，则代理签名的验证方程为 $g^T=R^R\left(K\widetilde{Y}\prod_{i=1}^{t}y_i\right)^{h(m,R)}\ \bmod p$。给定 m' 和 $\widetilde{Y}$，求解 R' 和 T' 的难度建立在单向函数 hash 和离散对数问题的难解性之上；同理给定 m'、R' 和 T'，可以得到满足上述方程的 $\widetilde{Y}$，但是在单向函数 hash 和离散对数问题的难解性假设基础上求解 K 和 $PGID$ 是困难的。

(3)最后来看一下 Hsu 方案提到的攻击——修改门限值 t。如果代理签名人合谋修改门限签名值，那么需要所有的代理签名人合谋才能做到这一点，在实际应用中意义不大。而且由(1)(2)(3)的分析可以知道，这种攻击在我们的改进方案中是无法实现的。

6 结论

代理签名提供了一种将签名权利分配给多人行使的方法，但是在签名过程中易受到合谋攻击。在改进方案中，我们改进了 Sun 的方案的这一安全漏洞：一旦受到合谋攻击，恶意的攻击者就可以伪造任意消息的签名。同时我们的改进方案也能够防止伪造攻击，保持了较高的效率。(收稿日期：2006 年 1 月)

参考文献

1.M Mambo,K Usuda,E Okamoto.Proxy Signature:Delegation of the Power to Sign Messages[C].In:IEICE Trans Fundamentals,E79-A:9,1996:1338~1353

2.M Mambo,K Usuda,E Okamoto.Proxy Signatures for delegating signing operation[C].In:Proc 3rd ACM Conference on Computer and Communications Security,ACM press,1996

3.S Kim,S Park,D Won.Proxy Signature,revisited[C].In:Proc of ICICS'97 International Conference on Information and Communication Security,1997:223~232

4.Zhang K.Threshold Proxy Signature Schemes.Information Security Workshop,Japan,1997:191~197

5.Sun Hung-Min,Lee N-Y,Hwang T.Threshold Proxy Signatures[J].IEE Proceedings-computers & Digital Techniques,1999;146(5):259~263

6.C-L Hsu,T-S Wu,T-C Wu.Improvement of threshold proxy signature scheme[J].Appl Math Comput,2003;136:315~321

7.Z-H Shao.Improvement of threshold proxy signature scheme[J].Computer Standards&Interfaces,2004;27:53~59

8.T Pedersen.Distributed provers with applications to undeniable signatures[C].In:Proc EUROCRYPT' 91,Lecture Notes in Computer Science,547,Springer,Berlin,1991:221~238

9.A Shamir.How to share a secret[J].Commun of the ACM,1997;22(11):612~613

(上接 107 页)

sactions on Computer Systems,1996;14(3):225~264

5.S A Brandt,L Xue,E L Miller et al.Efficient metadata management in large distributed file systems[C].In:Proceedings of the 20th IEEE/11th NASA Goddard Conference on Mass Storage Systems and Technologies,2003:290~298

6.Scott A Brandt,Ethan L Miller,Darrell D E Long et al.Efficient Metadata Management in Large Distributed Storage Systems[C].In:Proceedings of the 20th IEEE/11 th NASA Goddard Conference on Mass Storage Systems and Technologies,2003

7.G Ganger,B Worthington,Y Patt.The DiskSim Simulation Environment.http://www.pdl.cmu.edu /diskSim/index.html

8.F Wang,Q Xin,B Hong et al.File system workload analysis for large scale scientific computing applications[C].In:Proceedings of the 21st IEEE/12th NASA Goddard Conference on Mass Storage Systems and Technologies,College Park,MD,2004

分布式存储系统中元数据系统的研究与设计

作者： 苏勇， 周敬利， 姜明华， 刘钢， Su Yong， Zhou Jingli， Jiang Minghua， Liu Gang

作者单位： 苏勇,姜明华,Su Yong,Jiang Minghua(华中科技大学计算机科学与技术学院,武汉,430074;武汉科技学院计算机科学学院,武汉,430073)， 周敬利,刘钢,Zhou Jingli,Liu Gang(华中科技大学计算机科学与技术学院,武汉,430074)

刊名： 计算机工程与应用 ISTIC PKU

英文刊名： COMPUTER ENGINEERING AND APPLICATIONS

年，卷(期)： 2006，42(20)

被引用次数： 1次

参考文献(8条)

1. D Roselli. J Lorch. T Anderson A comparison of file system workloads 2000

2. P J Braam The Lustre storage architecture 2002

3. E Levy. A Silberschatz Distributed file systems:Concepts and examples 1990(04)

4. P F Corbett. D G Feitelso The Vesta parallel file system 1996(03)

5. S A Brandt. L Xue. E L Miller Efficient metadata management in large distributed file systems 2003

6. Scott A Brandt. Ethan L Miller. Darrell D E Long Efficient Metadata Management in Large Distributed Storage Systems 2003

7. G Ganger. B Worthington. Y Patt The DiskSim Simulation Environment

8. F Wang. Q Xin. B Hong File system workload analysis for large scale scientific computing applications 2004

相似文献(10条)

1. 学位论文 钟志勇 遥感影像信息系统若干关键技术的研究 2005

多时相、多谱段、多分辨率、多数据源遥感影像在多个领域发挥着作用，如环境监测、变化检测、目标识别与定位等等。建立遥感影像信息系统则是遥感数据在这些领域进一步开展应用的重要基础。本文针对当前遥感影像信息系统的特点和研究现状，重点对遥感影像信息系统的若干关键技术进行了系统、深入的研究，主要工作如下：

1.设计了一种通用的、支持分布式计算和具有典型三层计算结构的遥感影像信息系统结构模型，并据此建立了遥感影像数据库的层次对象数据模型。

2.在影像元数据有关研究的基础上，给出了一套遥感影像数据库元数据体系的设计。分析了海量数据物理存储系统设计的基本方法，提出了存储系统的抽象模型与性能分析方法和评价指标，并给出一个实现基于网络共享的分布式存储系统的方案。

3.引入了一个更符合人对于影像匹配的理解的广义影像匹配与配准的概念。利用多种匹配模式及金字塔影像多级匹配原则，有针对性地提出了一种新的匹配模式，改进了部分匹配算法的中间过程。设计了将遗传算法与小波多分辨分解和最小二乘影像匹配有机结合的快速、高精度的匹配算法。

4.根据人眼对于影像内容对准的认识，提出一个新的影像配准准则：配准度。该准则能够有效地衡量影像的配准程度，对影像的灰度属性没有苛刻的要求，并且能容忍影像中的一定量的噪声。设计了一种基于配准度和根均方误差的两步匹配算法，建立同名控制点对，并以迭代优化机制对控制点的位置进行优化，最后进行影像的精确配准。

5.将海量的、多尺度的遥感影像信息应用于三维可视化。对三维地形交互技术进行了探讨，通过纹理数据的LOD表示解决了大数据量景观模型的实时漫游问题，提出了化3维为2+1维的查询方法，并运用于三维查询中。

6.在对遥感影像信息系统的理论、技术、设计思想、实现方法全面系统研究的基础上，开发了一个实用、基于网络的实验系统并在多个项目中得到成功应用。

最后，关于进一步工作的方向进行了简要的讨论。

2. 期刊论文 石柯. 王庆春. 吴松. SHI Ke. Wang Qing-chun. WU Song 数据网格中一种基于副本和缓存的元数据管理系统 -计算机研究与发展2004,41(12)

元数据管理是数据网格的关键技术之一. 对全局分布式存储系统GDSS(global distributed storage system)中的元数据管理进行了改进,提出了一种基于副本和缓存的分布式元数据管理系统RCMMS(replication and cache based distributed metadata management system),缓存设置在GDSS系统中的存储服务点SSP(storage service point)端. 还讨论了RCMMS的设计、实现以及测试. RCMMS提供了动态管理元数据副本的有效算法. 分析和测试表明,副本结合缓存的元数据管理方案在性能上超过了GDSS现有的元数据管理系统,有着较好的可靠性.

3. 学位论文 刘群 基于可扩展对象的海量存储系统研究 2006

信息存储是人类社会永恒的需求。随着计算机技术的发展和应用的普及，信息存储容量成爆炸性地增长，现有网络存储系统已无法满足人们对于存储的需要。基于对象存储（Object-Based Storage，OBS）技术逢时崛起，利用现有的存储组件、处理技术和网络技术，通过简单方式来获得前所未有的高吞吐量，成为下一代网络存储的主流。它采用包含数据和属性的“对象”作为接口，既有了“块”接口的快速，又有“文件”接口的便于共享，并分离了存储数据的逻辑视图和物理视图，将存储数据的逻辑视图保留在元数据服务器中，而物理数据存放在基于对象存储设备（Object-Based Storage Device，OSD）中。同时，它将传统文件分解为系列数据对象，分发到一个或多个OSD 中。虽然对象给存储系统带来了一种新的理念，但现有的与对象相关的存储系统中对象都仅定义为非定长的数据单位，束缚了“对象”这个有着丰富内涵的词汇。

基于可扩展对象的海量存储系统（Based on Scalable Object Mass StorageSystem，BSO-MSS）吸取了OBS 的优点，在“对象”现有的含义基础上扩充，使它不仅只包括用户数据，还将目录、文件、存储设备管理等纳入对象之中，形成层次结构的对象体系结构，实现对象的分布存储、层次管理的模式，并建立基于存储对象统一访问模式，将块、对象和文件三种存储接口进行融合与统一。这样不仅具有统一逻辑视图、数据共享、主动服务、并行

访问、统一存储和易管理等特点，而且有着其他存储结构难以达到的高可扩展性和高性能。

通过建立系统广义随机Petri 网模型，对BSO-MSS 进行性能评价，模拟结果显示无论增加存储对象（Storage Object，SO）还是客户端，系统性能都随之增加。并采用测试工具iozone 对系统原型与Lustre 系统作对比测试，测试结果表明写性能超过Lustre，读性能略比Lustre 好，并验证了BSO-MSS 的广义随机Petri 网模型。

首次将存储系统与元胞自动机相结合，利用元胞自动机的原理，解析BSO-MSS动力演变规律。构建了一个通用框架的BSO-MSSCA 概念模型框架，并在此基础上，分析了两种具体元胞自动机模型。基于存储对象负载分配模型是将SO 解析为元胞，模拟了一个简单的负载均衡分配的动态变化，高度概括了BSO-MSS 的演变过程。

基于数据对象访问行为模型则分析数据对象的访问频率对系统的影响，结合数据对象访问的特征和主动性，通过机械学习适当调整数据对象的访问行为频率，使系统朝着稳定方向发展。通过分析基于存储对象的负载分配模型和基于数据对象的访问行为模型的演变过程，可以看出系统具有主动性、共享性、并行性、相关性等特性，是一个自组织管理的对象存储系统。

大规模分布式存储系统中，元数据高性能服务、负载均衡以及扩展性已成为一个重要的研究热点。在元数据服务器中，将元数据分解为目录对象和文件对象，目录对象为定位性元数据，提供文件所在位置和访问控制；文件对象为描述性元数据，描述文件的数据特性。每一个元数据服务器（Metadata Server，MDS）负责所有目录对象和自身的文件对象，这样充分利用MDS 中Cache，提高Cache 的命中率，减少磁盘I/O 次数，而且能够动态扩展MDS。同时，以目录对象ID 和文件名为关键字的哈希值作为局部元数据查找表（Local Metadata Lookup Table，LMLT）的索引，获得相应的MDS_ID。一旦目录权限改变、更名、移动目录、修改权限等都不会造成元数据的迁移。通过Bloom Filter 算法将每个MDS 的LMLT 压缩成一个摘要，能够实现快速的元数据查找。同时采用主从备三重链式结构的MDS 服务，不仅在未提高硬件成本下能够保证系统高可靠性和可用性，而且根据热点访问进行迁移，实现负载均衡。SO 是BSO-MSS 重要组成单位，它与OSD 不同之处是本身具有“接口”与“状态”标识，由数据、属性和方法组成，这样对现有的T10 OSD 标准进行了扩充。由于数据对象是通常在一维空间中命名，传统文件系统管理大量数据对象的效率是极其低，采用线性哈希查找算法，由负载因子控制分裂和合并，与传统文件系统的树结构查找相比，哈希法查找时间复杂度为O(1)。同时，针对Ext2 文件系统中数据访问至少需两次以上的磁盘操作特性，将数据的块地址和长度链接在一起，作为对象的扩展属性，连同数据对象一起存储到磁盘中，这样无论数据对象大小为多少，磁盘访问次数仅为两次。在BSO-MSS 中，负载与众多因素相关，如请求队列长度、CPU 处理能力、内存大小、网络带宽、磁盘带宽和磁盘容量等。负载柔性放置策略不仅考虑网络的影响，而且考虑SO 之间存在差异，并设置权重，权重大的SO 担负较多的负载。依据SO 属性中信息统计出负载特征，以系统响应时间为代价，自适应选择SO 数目，采用不同大小的分条进行存储，使BSO-MSS 具有更高的性能、可扩展性和自适应负载均衡能力。

4.期刊论文 郭莉.刘伟.黄海.GUO LI.LIU WEI.HUANG HAI 大型分布式存储系统高效元数据管理 -微计算机信息

2008,24(9)

在大型分布式存储系统中,高效元数据管理是保证整个系统运行的关键环节.目录子树和纯哈希方法是元数据管理采用的一般技术,但这两种技术在高速频繁的并行数据访问情况下会导致瓶颈.本文主要阐述一种元数据管理技术-懒散混合法(Lazy Hybrid,LH),这种技术对以上两种技术进行综合,取长补短,效果明显.

5.学位论文 靳超 主动存储系统结构的研究 2003

随着计算机技术的发展以及用户对于存储需求的日益增长，主动存储系统成为热点研究。如何利用未来存储设备上的计算能力来支持高性能的计算和高效的存储访问成为问题的关键。本文首先提出了存储对象和应用对象相结合的设备访问接口结构，并提出了一种主动存储系统的可扩展模型，该模型有效地利用设备的物理特征来提高设备访问性能，在该模型的基础上提出了存储管理系统中优化服务质量的算法。在这些研究的基础上，设计并实现了一个基于主动存储设备的分布式存储系统，取得了较好的效果。

本文的贡献主要包括以下几个方面：(1)提出了一种主动存储体系结构的计算和存储访问模型，同时研究和分析了该模型相对于传统存储系统的性能优势，并探讨了适用于该计算模型的应用实例。实验结果表明基于该模型所完成的并行数据敏感性应用任务在性能方面得到了显著提高。

(2)提出了一种在主动存储设备上构建分布式存储系统的可扩展元数据访问机制。基于该机制构建的分布式存储系统能够保证系统的元数据访问具有良好的可扩展性能。实验数据表明，使用该机制的系统，随存储结点的增加吞吐率呈近似线性扩展。

(3)提出了一种利用主动存储设备上的计算能力通过理解设备的物理特征来提高I/O访问性能的机制。利用本方法可以针对不同访问模式的数据进行相应的优化处理。实验数据表明，对于同步的I/O写请求，本方法能够将性能提高5-8倍左右。

(4)提出了一种保证服务质量(QoS)的虚拟存储空间管理算法。该算法能够针对用户的不同级别和访问要求支持相应的服务质量。在充分利用整体资源的条件下，对于高优先权的用户分配较多的存储资源；同时尽可能地保证低优先权用户的权益。

(5)设计了一个主动存储设备(TOSD)。该设备采用了本文中的部分研究成果，支持高效的、事务的并发访问。并且基于该主动设备完成了一个分布式存储系统——TODS，该系统的特点是提供透明持久化的对象式访问接口，能够提供便捷的、高效的分布式数据存储功能。同时在该系统的基础上开发了一个可扩展的Email服务应用。

6.期刊论文 钱薇.李贤君.Qian Wei.Li Xianjun 基于iSCSI 的对象存储系统的研究与设计 -计算机与数字工程

2009,37(7)

对象存储系统已经成为分布式存储系统领域的研究热点,它在可扩展性、安全性、高效性以及性能上较传统的SAN和NAS等存储体系结构有很大改进.采用较为成熟的iSCSI架构,可以开发出实用的iSCSI对象存储系统.

7.学位论文 罗睿 遥感图像信息系统的设计与分析 2001

该文主要工作体现在如下几个方面:1、全面地总结、分析了Web信息服务的实现方法、Web应用和传统应用的不同特点以及一般Web数据库应用系统的基本结构.2、提出并实现了一种通用的、支持分布式计算和具有典型三层计算结构的遥感图像信息系统软件结构模型.3、建立了遥感图像数据库的层次对象数据模型,提出了对象-关系模型转换的具体方法.4、完成了一磁疗遥感图像数据库元数据体系的设计.5、分析了海量数据物理存储系统设计的基本方法,介出了存储系统的抽象模型与性能分析方法和评价指标,并给出了一个实现基于网络共享的分布式存储系统的具体方案.6、针对遥感图像无损压缩研究,对整数小波变换的去相关能力、边界处理、K系数处理、系数统计特性进行了系统深入的实验研究.7、对两类常用的基于整数小波变换的无损压缩方法进行了有针对性的完善,提出了一种更加符合大型遥感图像网络应用的无损压缩方法.8、提出了一种能够把图像关系查询和基于内容查询相统一的集合代数模型.系统地研究了GIS支持下遥感图像基于内容查询.9、探讨了图像划询语言的设计思想和语言的结构形式,建立了一个基本的图像查询语言系统.10、开发了一个实用、基于Web技术的完整系统——Image Map Library,并在多个项目中得到成功应用.

8.期刊论文 刘群.冯丹.Liu Qun.Feng Dan 基于层次结构的元数据动态管理方法的研究 -计算机研究与发展

2009,46(z2)

在大规模分布式存储系统中,元数据高性能服务和扩展性已成为一个重要的研究热点.在元数据服务器(metadata server,MDS)中,将元数据分解为目录对象和文件对象.目录对象为定位性元数据,提供文件所在位置和访问控制;文件对象为描述性元数据,描述文件的数据特性.每个MDS负责所有目录对象和自身的文件对象,同时,以目录对象ID和文件名为关键字的Hash值作为局部元数据查找表的索引,通过Bloom Filter算法将每个MDS的局部元数据查找表压缩成一个摘要,这样既可利用MDS中Cache,提高Cache的命中率,减少磁盘I/O次数,动态扩展MDS,又能够实现快速的元数据查找.

9.学位论文 赵辉 麒麟天机存储系统的负载均衡与副本一致性技术研究与实现 2008

麒麟天机安全存储系统是由国防科学技术大学计算机学院研制的基于银河麒麟操作系统的安全存储系统，数据集中加密存储于服务器，由服务器操作系统实现内核级的透明加解密，支持基于保险箱的文件加密与安全共享。

麒麟天机安全存储系统目前采用单服务器的集中数据存储方式，难以满足大用户量的数据访问需求。因此，拟摸索一种系统扩展方案。

麒麟天机安全存储系统采用samba[1]作为数据输出子系统，因此，本文认为研究部署高性能samba分布式文件系统是系统扩展的第一步。目前，samba分布式文件系统由一个称为根服务器的元数据服务器和很多后端存储服务器组成，根服务器向用户提供统一的全局虚拟视图，它以做符号链接的方式映射后端存储服务器上的物理存储资源，一个链接可以对应多个链接目标，每个目标作为一个冗余副本。客户端访问文件系统时，根服务器向客户

端返回数据存储服务器的符号链接目标列表，客户端根据获得的列表，按顺序访问数据存储服务器，直到数据访问成功或者全部访问失败为止。因此，在大部分情况下，所有的访问会集中到某个数据存储服务器上，造成服务器之间的负载严重失衡。同时由于麒麟天机安全存储系统为用户提供透明加密存储服务，数据的修改和更新是基本功能，就会造成分布式存储系统上多个副本问的数据不一致问题，而samba现有的机制并没有保证这种多副本数据的一致性。论文的目的就是研究samba分布式文件系统的负载均衡和副本数据一致性问题。

对于负载均衡问题，本文提出一种基于后端服务器性能的负载均衡策略。该策略主要由两个子策略组成，即后端负载反馈策略和前端负载调度策略。后端存储服务器周期性地反馈负载指标，该指标用反馈周期内服务器网卡平均每秒的流量表示。前端根服务器依据服务器的峰值负载指标和动态反馈的当前负载指标计算出服务器当前剩余负载指标，将其作为服务器的性能指标，进行负载调度。在本系统中具体体现在根服务器在向客户端返回符号链接目标列表前根据目标服务器的当前剩余负载指标从大到小对列表进行重新排序。

对于副本数据一致问题，本文的策略是：一、只允许在主副本上进行更新，可以通过将所有具有更新权限的用户定位到相同副本的方式来保证，这样就解决了写写冲突造成的不一致性。二、对于读写冲突造成的不一致，本文采用主动推式更新传播的策略，即主副本更新后会将更新主动推入其他的副本，使多个副本达到一致。

通过netbench对本文的改进后的系统进行测试，发现整个系统的吞吐率和对用户的平均响应时间都得到了很大幅度的提升。

10.期刊论文 万继光.詹玲.WAN Ji-guang.ZHAN Ling 集群多媒体存储系统的两级元数据管理 -小型微型计算机系统2009,30(4)

随着网络上多媒体数据的爆炸性增长,对海量可扩展的存储系统的需求也快速增长.CMSS(Cluster Multimedia Storage System)项目采用分布式存储系统结构:一种自治的高性能的基于PC的存储集群系统.CMSS采用两级的元数据服务器结构,通过分离存储数据的逻辑视图与物理视图,全局逻辑视图由专用的全局元数据服务器来管理,局部逻辑视图和物理视图由各个存储服务器上的本地元数据服务器来管理.在详细介绍了CMSS系统两级元数据管理方案的同时,进行了相应的试验测试和性能分析.

引证文献(1条)

1.刘祖云.胡进德 分布式共享存储研究[期刊论文]-成都大学学报（自然科学版） 2008(1)