

基于对象的进化存储系统研究^{*})

董晓明 谢长生

(华中科技大学计算机学院 信息存储系统教育部重点实验室 武汉 430074)

摘要 存储管理的成本目前已经超过了硬件成本。本文给出了一种新的存储结构,其中包含了自动化管理的功能以简化存储管理。进化存储系统的概念是指实现动态平衡和优化的基于磁盘的存储,其失效部件将会被热备份部件立即更换(物理进化),其数据布局 and 分布会根据新部件的特性进行自动重组(逻辑进化)。同时,我们还描述了称为 SMART 的实现分散进化存储系统的目标和可行技术方案,包括自安全、自维护、自适应、自恢复和自测试。我们认为进化存储系统应该是基于对象的,通过智能存储设备达到更好的可用性和可维护性。

关键词 磁盘,进化存储系统,自适应,对象存储

A Study on the Object-based Evolving Storage Systems

DONG Xiao-Ming XIE Chang-Sheng

(Key Laboratory of Data Storage System, School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074)

Abstract Storage management costs now exceed storage hardware costs, and this paper outlines a storage architecture that integrates automated management functions to simplify them. Evolving Storage System (ESS) is the concept of a dynamically balanced and optimized disk-based storage, in which failure component should be replaced immediately by another hot spare component (physical evolving); and the data layout and distribution should be reorganized automatically according to the attributes of new components (logical evolving). We also describe the feasible SMART techniques toward ESS, including self-securing, self-maintaining, self-adaptive, self-recovering, and self-testing. We believe that it's reasonable for ESS to be object-based internally, aiming better availability and maintainability by intelligent storage devices.

Keywords Disks, ESS, Self-adaptive, Object storage

1 引言

大规模存储系统通常由专用磁盘阵列控制器、光纤通道存储区域网(FC-SAN)、自动磁带库等构成,非常复杂、昂贵。目前已经出现了向分布式异构环境下的通用部件(例如 PC)转移的趋势。使用日常唾手可得的硬件和软件构建系统可以大大降低每 TB 存储的成本。同时,用于存储管理的人力资源成本达到软硬件成本的 4~8 倍,因而人们更加关注系统的总体拥有成本(TCO)而不单单是购买价格。

本文描述了进化存储系统(Evolving Storage System, ESS)的轮廓。ESS 是基于磁盘的存储系统,它通过替换存储部件来建立保持动态平衡和优化的状态。通过实时监控每个部件,失效部件会立即被备份部件取代。数据布局 and 分布状况是自适应的。因此,可以得到一个高可用和易于维护的存储系统。

我们还描述了实现 ESS 的五项可行技术,包括自安全(self-securing)、自维护(self-maintaining)、自适应(self-adaptive)、自恢复(self-recovering)和自测试(self-testing),同时简称为 SMART,代表智能存储设备。

为了达到某种程度的智能,必然需要为存储设备和接口设置更多的属性,因此有必要利用对象来进行更好的存储管

理。对象可以构成基于属性的认知环境,使存储设备可以感知当前环境和用户应用,从而更好地分配资源。

本文第 2 节介绍了本文的研究背景和相关领域的发展趋势;第 3 节给出了进化存储系统的概念和目标;而自适应的数据分布作为其中一项关键技术,在第 4 节进行讨论;第 5 节讨论了基于对象的存储;最后全文总结。

2 研究背景和趋势

磁盘驱动器是最重要的外存储部件。我们先介绍一些有关背景和发展趋势。

硬件发展符合摩尔定律 现代计算技术的发展趋势在存储、处理和网络三个方面分别遵循 Moore's, Amdahl's 和 Gilder's 定律^[1]。存储系统传统的评价指标是成本、容量、带宽和延迟时间。

容量:存储设备的容量已经在按照摩尔定律增长。特别地,磁盘容量增长显著(每 18 个月翻一番),今天已经达到 400GB 左右。

带宽:同样,磁盘传输率也有很大增长,不过没有容量增长那么快(大约每 3 年翻一番)。磁盘的容量/带宽之比平均每十年增加十倍。

延迟时间:磁盘寻道和旋转延迟的改进比带宽更缓慢。

^{*} 本课题得到国家自然科学基金资助(项目编号:60273073 和 60303031)。董晓明 博士研究生,主要研究方向为:网络存储技术,嵌入式系统。谢长生 教授,博士生导师,主要研究方向为:计算机系统结构,网络存储系统,超高密度光存储技术,存储安全和存储备份技术。

事实上,网络延迟已经可以与光速相比,而磁盘随机寻道的延迟在 9ms 左右(有可能在 20 年内缩短到 1.7ms)。

磁盘正在替代磁带 磁带容量会比磁带速度增长更快,而存取时间会基本保持不变,这会造成更严重的存取问题,可能需要几天的时间来扫描磁带存档。历史上,磁带、磁盘和内存曾经保持 1:10:1000 的价格比,现在这个比值应该是 1:3:300。值得注意的是磁盘价格正在接近近线存储(磁带)的价格。

系统管理员会发现,备份 1TB 到磁带没有问题,但是要从磁带恢复 1TB 的数据需要很长的时间。随着存储规模逐渐达到 PB 甚至更多,管理员正在改变策略,维护数据的多个在线磁盘备份从而不必从磁带恢复数据。

磁盘基本上正在替代磁带作为备份设备,近线磁带库仅仅作为归档使用。

网络作为 I/O 背板 一个有趣的现象是,网络带宽正在超过磁盘带宽。事实上,千兆(Gigabit)以太网已经得到广泛使用。今天的光纤传输带宽已经达到 40Gbps。可以预见的是,理想的存储系统应该通过网络存储数据分条(stripe)到许多磁盘上,聚合多个磁盘带宽以达到网络带宽。

通过向网络添加更多的磁盘,可以减轻磁盘寻道时间和带宽所带来的问题。网络带宽和 CPU 处理能力将决定分布式存储系统的可用带宽。

管理变得更加重要 随着存储设备的价格不断降低,现在存储管理成本已经超过了硬件成本(类似地,PC 的管理成本超过了硬件成本)。有个说法是一个人只能管理价值一百万美元的磁盘存储,在今天也就是 30TB。

数据存储规模会越来越大。我们面临的巨大挑战是简化存储管理和访问。一个合理的方式是使数据组织、存取和保护等各项任务自动化。

3 进化存储系统

这里我们将介绍进化存储系统(ESS)的概念和目标。

3.1 概念

大规模存储服务的需求正在迅速增长。可扩展计算、多媒体和虚拟化带来数字化信息爆炸性增长,数据的重要性和安全性日益增加,24×7 的服务需求要求存储系统具有极高的可用性和可维护性。迎接这种挑战的主要技术是高性能磁盘阵列、附网存储(NAS)、存储区域网(SAN)、IP 存储技术和存储管理软件等。

但是现有技术仍面临很多问题,比如:元部件(磁盘等)的飞速发展存储系统整体结构相对固定的矛盾,数据组织和分布状态不能适应变化的 I/O 请求的矛盾,相对固定 RAID 级别与多样性应用之间的矛盾等。

第一,目前的系统有一个普遍存在的问题,即元部件的飞速发展和整体结构相对固定的矛盾。以磁盘阵列技术为例,它的构成元部件是磁盘,一旦某个磁盘出故障,它会在备份的新盘上重构丢失的数据。然而,无论新盘在速度和容量上比其它旧盘高出多少倍,重构后的磁盘阵列总体性能不会有丝毫提高。目前的存储系统不会由于更新了部件而得到整体性能的提高。因而,随着技术的进步,这种性能停滞不前的系统很快就会过时,用户不得不购买新的系统来满足新的需求。

第二,数据固定的组织方式使得数据分布不适应变化,系统整体性能呈现一种随使用而退化的趋势。例如,磁盘的性能随着反复改写的次数增多而性能逐步下降,这是众所周知

的碎片问题。

第三,系统结构的组织方式较为固定,不能适应应用的多样性和动态性。例如,磁盘阵列的级别一般设定好后就固定下来,难以适应各种不同性质的应用,如对事务处理设定了 RAID5 而具有较高的性能,但对流媒体的应用就不大合适。

我们认为上述问题存在的原因在于现有存储系统物理和逻辑的组织是一种静态的结构,而静态组织结构模型不能很好地刻画处于不断变化之中的系统。这种结构往往适合于特定的应用需要,而且缺少适应不断变化的存储要求的机制。

“进化存储系统”的概念就是为了解决这些问题而提出来的,其关键思想是:进化存储系统是遵循优胜劣汰原则,由软硬件构件按一定结构组成的动态存储系统。系统实时监测构件状态,一旦发现工作构件处于不良状态或有向更好状态迁移的可能,就用备用构件构造出一个具有同样功能但处于更好状态的构件代之。原构件或被再生为备用构件,或被淘汰,从而使整体性能得到进化。构件可以是磁盘、磁盘阵列及互联设备,也可能是某种功能的软件。状态可以是物理状态,也可以是数据分布状态。

这种进化存储系统从整体上分为两种类型的进化,一是物理进化,例如用更高的性能和更大容量的新磁盘取代故障磁盘,就是一种物理进化。这个过程可以在线完成,不但不影响系统内部数据的可用性,而且还应该随着部件性能提高而获得整体性能的提升。二是逻辑组织结构进化,即磁盘系统中数据的组织以及整个系统中各种不同属性数据的分布可以随着外在的数据流 I/O 的变化而相应改变,以期获得最好的系统性能。例如不合适的 RAID 级别可进化为合适的 RAID 级别。

进化存储系统具有容量和性能随时间进化的特点,这是一种能在进化过程中保持动态平衡的系统。它呈现给用户的外在特征是系统越用越好,不仅使前述问题迎刃而解,而且具有极高的可用性和易管理性,是一种新的存储系统概念。它满足了现代企业级存储系统对性能、容量、可用性、可维护性和可扩展性等多方面的要求,具有十分广阔的应用前景。

3.2 目标:SMART

大规模存储系统包括大量可配置的共享资源和 I/O 网络,这无疑使管理任务的工作量和复杂性增加,甚至手工管理不再可行。为了解决这些问题,我们认为存储系统必须变“聪明”,具有自安全、自维护、自适应、自恢复和自测试的能力。下面我们分别描述这几个方面的技术。

3.2.1 Self-securing 数据安全是存储系统最重要的问题之一。存储系统通常依赖于主机操作系统和入侵检测系统(IDS),当检测到并终止非法入侵之后,系统管理员将面对两项艰巨任务:确定入侵造成的损害,然后从备份系统中恢复数据。我们更需要一个自安全的存储设备,能够作为一个独立的实体运行,并保证数据安全。

S4 项目实现了一个自安全的存储服务器^[2],可以防止入侵者篡改或者永久删除数据。既然不能区分入侵者和合法用户,S4 认为所有请求都是可疑的。存储设备内部将所有版本的数据和请求保留一段时间(比如一周或者一个月),从而提供给管理员一个时间窗来发现和消除入侵。S4 使用类似日志文件系统(LFS)的日志结构保存不同版本的数据,并维护一个只能追加记录的请求审核日志。

3.2.2 Self-maintaining 对大规模磁盘存储系统来说,维护是个大问题。随着价格的下降,如今存储管理的成本已

经超过了硬件成本。最关键的是,维护不当可能造成数据的丢失。

对系统管理员来说,当然希望系统能够始终正常运行(哪怕管理员不在)。然而,所有的系统都需要人工干预,例如更换硬件,因此没有系统能够实现完全自动化(完全不需要管理员)。“自维护”的意思是说存储系统能够适当减轻、简化管理员的工作:1)仅需要定时维护;2)在进行维护的时候,需要什么已经明确指明了。

除了降低成本之外,这样的维护还有助于减少操作人员出错的机会。构建一个自维护的系统需要下列部分:1)系统必须具有一定数量的冗余,没有单点失效;2)另外,正确标识失效和依赖关系,使系统尽量自治;3)最后是修复。

Tertiary 磁盘项目给出了一个“自维护”的方案^[3]。该项目是由交换以太网连接的 PC 节点集群构成的大规模磁盘阵列系统。

3.2.3 Self-adaptive 逻辑组织结构的进化是进化存储系统需要研究解决的关键问题之一,包括大量磁盘的组织结构、I/O 和数据特征分析、自适应数据分布策略等方面。而自适应是逻辑进化的基础。自适应的目的是让存储系统自动调整来适应环境的变化,期望存储系统在存储构件(比如磁盘和网络存储节点)加入和离开的过程中有稳定的性能和可用性。

本文第 4 节更详细讨论了自适应数据分布的有关问题。

3.2.4 Self-recovering 系统中总会有失效部件,我们必须研究如何修复失效部件并尽快恢复数据。

面向恢复的计算(Recovery Oriented Computing: ROC^[4,5])认为硬件故障、软件 bug 和操作人员错误都是需要正视的事实,而不是需要解决的问题。ROC 关注“平均修复时间(MTTR)”而不是平均故障时间(MTTF),致力于缩短修复时间从而提供高可用性。既然大部分系统维护工作是处理失效,ROC 同时降低了 TCO(总体拥有成本)。Shimon Peres 说,“如果一个问题没有解决之道,那么它可能根本不是问题而是事实,不需要去解决只需要去面对。”这是 ROC 的信条。看上去这个“Peres 定律”的重要性很快会超过摩尔定律。

ROC 系统要求硬件和数据的冗余。为了达到快速发现失效的目的,ROC 应该在构件和系统级进行广泛的自测试和检测。为了辅助诊断,应该跟踪所有构件的状态,还应该具有引入仿真故障并触发恢复的机制。

3.2.5 Self-testing 自测试包括两个方面:其一是实时监测和诊断存储构件,包括硬件(比如磁盘驱动器、控制器和节点网络接口等)和分布式软件本身。

自测试的功能包括通常的监测以发现潜在问题,例如故障节点、性能瓶颈和拒绝服务(DOS)攻击。像嵌入式系统的内建自测一样,一旦扫描发现潜在的错误,就立即重新启动该部件。

其二是在线析取磁盘驱动器和其他部件的性能特征参数。自动获得准确的特征信息非常重要,比如容量(占用和空闲空间)、驱动器和节点带宽等,这些信息将会影响数据放置和迁移。文[6]描述了实验方法析取磁盘参数而不要求预先知道相关信息。

4 自适应数据分布

这里我们将讨论磁盘的数据组织和分布问题。对数据分布的研究可以位于不同的层次(如图 1):(1)一个磁盘上的数据布局;(2)一个主机内多个磁盘之间的数据分布;(3)网络互

联的(分散的)多个节点之间的数据分布。

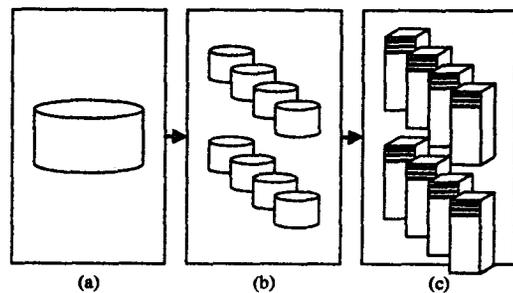


图 1 研究数据分布的三个层次:(a)磁盘内部;(b)多个磁盘之间(主机内部);(c)分散主机节点之间

4.1 磁盘数据布局

磁盘布局是存储的逻辑视图与物理磁盘位置之间的映射。为了提高磁盘 I/O 性能、减小访问时间,早期的研究主要关注磁盘数据块的布局、I/O 调度等,通过减少寻道时间来达到目的。数据布局的基本准则是将经常访问的数据块聚集在一起。

Organ pipe heuristic^[7]是有效的数据放置策略,其基本思想是将最经常访问的数据放置在磁盘中间(的磁道),次经常的数据靠中间两侧放置,直到最不经常访问的数据放置在磁盘边缘。研究^[8]表明,依据该策略重新组织磁盘磁道可以将寻道时间缩短 40%~45%。

磁盘 shuffling^[9]以及该领域的其它研究也反映了类似的结果,或者是针对特定的应用,如视频点播(VOD)进行磁盘数据的重新分布^[10,11]。

4.2 主机内多磁盘数据分布

磁盘阵列(RAID^[12,13])是今天企业级存储系统的标准解决方案。RAID 结构的原理很简单:用高速互连媒介聚合任意多的慢速存储设备成为一个较快速的逻辑存储服务,而所有的应用并没有感知到这种聚合。最常用的是 RAID 级别 0(分条, stripe), 1(镜像, mirror)和 5(块奇偶校验的分条)。RAID 将一个分条的数据分为 N-1 个数据块和 1 个奇偶校验块(数据块对应的异或)。每个数据块和校验块保存在不同的磁盘上。RAID 由多个磁盘的并行提供高带宽,同时由奇偶校验提供容错。

但是 RAID 有三个局限性:糟糕的小写性能,昂贵的专用硬件(控制器),以及有限的扩展性。准确配置 RAID 以满足负载要求、充分发挥 RAID 性能是件困难的工作。管理员必须深入理解各种 RAID 级别和不同工作负载的特点。另外,为了充分利用新增磁盘(容量更大、速度更快)的优势,管理员不得不重新创建 RAID、重新格式化、重新在各磁盘分区上分布数据。

HP AutoRAID^[14]建造了一个两级层次存储子系统,由 RAID-1 和 RAID-5 构成。AutoRAID 结合了镜像的性能优势和 RAID-5 的容量优势,将活动数据存放在 RAID-1 上,而相对不活动的数据存放在 RAID-5 上。但是 AutoRAID 也有其局限性,即要求整个存储系统只能有部分数据处于活动状态,且活动子集的改变相对较慢。

4.3 分散数据放置

集中化的文件系统从根本上制约了性能和可用性,商业系统越来越多地使用专用存储区域网(如基于光纤通道 SAN)来互连存储设备以及服务器,添加更多存储设备到网络

就可以扩展带宽和容量。

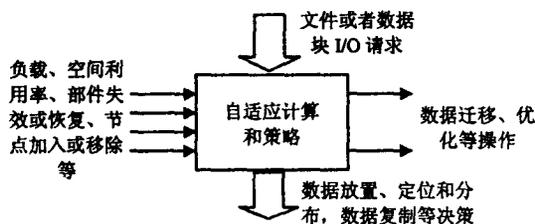


图2 一种可行的自适应数据分布结构

创建和部署如此大规模分散存储系统的最大障碍是其可管理性。当然，所有的系统都离不开管理、维护；而分散特性更增加了维护的难度，也更容易出现失误。在不久的将来，人们会更加关注可管理性而不是性能。图2显示了一种可行的结构。

自组织 在 Sorrento^[15] 项目中，数据定位方案提供了不受限制地将数据段放置到任意存储服务器的灵活性。数据段不会被限制在它初始创建的位置，系统动态地调整其位置来平衡 I/O 负载、平衡存储空间使用率。

在添加和减少存储节点时，存储集群最好能够保持持续不变的性能和可用性。我们称能够适应集群节点变化的存储集群为“自组织的(Self-organizing)”，其目标是使系统自动适应集群环境的变化。

“自省”和免维护 OceanStore^[16,17] 的目的是通过一个全球规模的免维护(Maintenance-free)的基础结构，满足基于Internet的分布式系统的存储需求。数据被复制并分散到不同地理位置，以避免硬件失效、配置错误和自然灾害带来的数据丢失。

OceanStore 项目将会由数以百万的服务器组成，具有不同的连接、磁盘容量和计算能力。服务器和设备会连接、丢掉连接或者偶尔失效。手工调节如此大规模、如此多样性的系统，其复杂性是惊人的。为了解决这个问题，OceanStore 使用“自省(Introspection)”——一种模仿生物系统自适应性的结构。自省机制由监测和优化模块组成，监测模块监视系统运行的状态并保存动作的历史记录，同时对监测结果进行分析并提取出操作模式，然后优化模块根据这些结果对系统进行调节。

5 基于对象的存储

存储系统规模持续增长，管理这些资源并满足性能需求成为一个巨大挑战。我们认为自动化的、基于对象的存储是这个问题的解决之道。

当前的存储设备接口(SCSI 和 ATA/IDE)应该具备更多的功能，应该从这种无智能、外部管理的状况向智能、自我管理发展，存储设备应该意识到其上层应用的情况。这种智能设备需要更复杂的接口，我们认为这个未来的接口应该基于存储对象。

如文^[18]所述，一个存储对象是存储设备上的一组字节，用通常的方法访问，属性描述数据的特征，安全策略防止未授权的访问。对象可以看作是两种技术的结合：文件和数据块。对象是一种存储容器，具有类似文件的高级接口，由数据块组成，具有用户可访问的属性和设备管理的元数据。

“属性管理的存储”项目^[19]研究自管理的存储系统，通过声明应用的需求和设备的特征来实现自动管理机制。该系统包括映射引擎，可以自动将对象映射到设备，同时保证以最低的成本满足性能需求。对象可以透明地迁移，以实现容错、性

能改进和分级存储管理。

Self-*^[20,21]项目的目标是自组织、自配置、自调节、自修复和自管理的系统，由网络连接的智能“砖块”构成，每个砖块提供必要的资源(包括 CPU 和 RAM)以缓冲和组织数据。“工人”是那些自适应地调节自己的砖块，路由器负责将请求转发给合适的工人。该项目实现的第一个原型 Ursa Minor 就是基于对象的存储系统。

随着对象存储的浮现，存储设备有必要主动学习其运行环境的重要特征。基于数据块的存储设备大多不了解使用存储的用户和应用。而基于对象的存储设备能够理解数据块之间的关系，并利用这些信息更好地组织数据。

总结 我们认为可用性和易用性是现代存储系统最重要的指标。进化存储系统是一条可行的应对挑战的途径，它是智能化的、基于对象的。对象存储可以充分发挥设备的能力，更进一步地理解应用并促使设备的智能化。

参考文献

- 1 Gray J, Shenoy P. Rules of thumb in data engineering. In: Proc. of the IEEE Intl. Conf. on Data Engineering, San Diego, CA, 2000. 3~12
- 2 Strunk J D, et al. Self-Securing Storage: Protecting Data in Compromised Systems. In: Proc. of the 4th Symposium on Operating Systems Design and Implementation, 2000
- 3 Asami S, Talagala N, Patterson D. Designing a Self Maintaining Storage System. In: Proc. of the 1999 IEEE Symposium on Mass Storage Systems, 1999
- 4 Patterson D A, Brown A, Broadwell P, et al. Recovery-Oriented Computing (ROC): Motivation, Definition, Techniques, and Case Studies; [UC Berkeley Computer Science Technical Report UCB/CSD-02-1175]. March, 2002
- 5 Brown A, Patterson D A. Embracing Failure: A Case for Recovery-Oriented Computing (ROC). In: 2001 High Performance Transaction Processing Symposium, Asilomar, CA, Oct. 2001
- 6 Talagala N, Arpacı-Dusseau R H, Patterson D. Microbenchmark-based extraction of local and global disk characteristics; [Technical Report CSD-99-1063]. University of California, Berkeley, 1999
- 7 Wong C K. Minimizing Expected Head Movement in One-Dimensional and Two-Dimensional Mass Storage Systems. Computing Surveys, 1980, 12(2)
- 8 Vongsathorn P, Carson S D. A System for Adaptive Disk Rearrangement. Software-Practice and Experience, 1990, 20(3)
- 9 Ruemmler C, Wilkes J. Disk Shuffling; [Technical report HPL-91-156]. Hewlett-Packard Company, Palo Alto, CA, Oct. 1991
- 10 Akyurek S, Salem K. Adaptive block rearrangement. ACM Transactions on Computer Systems, ACM Press, 1995, 13(2): 89~121
- 11 Sarhan N J, Das C R. Adaptive Block Rearrangement Algorithms for Video-On-Demand Servers. In: Proc. of the 2001 Intl. Conf. on Parallel Processing, IEEE Computer Society, 2001. 452~462
- 12 Patterson D A, Gibson G A, Katz R H. A case for redundant arrays of inexpensive disks (RAID). In: Proc. of the Intl. Conf. on Management of Data (SIGMOD), 1988
- 13 Chen P M, Lee E K, Gibson G A, et al. RAID: high-performance, reliable secondary storage. ACM Computing Surveys, 1994, 26(2): 145~188
- 14 Wilkes J, Golding R, Staelin C, Sullivan T. The HP AutoRAID hierarchical storage system. ACM Transactions on Computer Systems, 1996, 14(1): 108~136
- 15 Tang H, Gulbeden A, Zhou J, Chu L, Yang T. Sorrento: A Self-Organizing Storage Cluster for Parallel Data-Intensive Applications; [Technical Report 2003-30]. UCSB, Sep. 2003
- 16 Kubiatowicz J, Bindel D, Chen Y, et al. OceanStore: An architecture for global-scale persistent storage. In: Proc. of the Intl. Conf. on Architectural Support for Programming Languages and Operating Systems (ASPLoS 00), ACM, 2000
- 17 Rhea S, Wells C, Eaton P, et al. Maintenance-free global data storage. IEEE Internet Computing, 2001, 5(5): 40~49
- 18 Mesnier M, Ganger G R, Riedel E. Object-based Storage. IEEE Communications Magazine, 2003, 41(8): 84~90
- 19 Golding R, Shriver E, Sullivan T, Wilkes J. Attribute-managed storage. In: Workshop on Modeling and Specification of I/O, San Antonio, TX, Oct. 1995
- 20 Ganger G R, Strunk J D, Klosterman A J. Self-* Storage: Brick-based Storage with Automated Administration; [Carnegie Mellon University Technical Report, CMU-CS-03-178]. Aug. 2003
- 21 Strunk J D, Ganger G R. A Human Organization Analogy for Self-* Systems. In: First Workshop on Algorithms and Architectures for Self-Managing Systems, San Diego, CA, June, 2003