

一种高可用对象存储系统的数据组织研究

詹玲¹ 张强善^{2,3} 万继光¹

(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 武汉光电国家实验室 武汉 430074)¹
(武汉大学计算机学院 武汉 430072)² (信阳职业技术学院网络中心 信阳 464000)³

摘要 通过对现有不同存储系统的容错能力进行认真分析,提出了一种新的高可用对象存储系统架构 HAOSS (High Availability Object Storage System)。HAOSS 系统分两层:上层存储的对象采用在多个设备之间的数据备份的方式来实现高可靠性,多个备份能够同时对外提供服务,保证系统的高性能,但是磁盘利用率比较低。下层采用 RAID5, RAID6 以及 RAID_Blaum 等不同的容错编码来实现多盘容错功能,磁盘利用率较高,但是由于编码越来越复杂,容错编码需要大量的计算,性能受到很大的影响。在数据组织上,新对象和热点对象放上层,这样大部分请求都能在上层命中,从而保证系统的性能。下层主要用来放不常用的数据,磁盘利用率比较高。在 1000Mbps 以太网环境下对 HAOSS 性能进行了测试,结果表明,HAOSS 顺序读写性能都比较好,最大可以达到 104MB/s,达到 1000Mb 以太网的理论最大物理带宽。

关键词 数据组织,高可用,对象存储,容错

中图法分类号 TP303 **文献标识码** A

Data Organization Research of the High Availability Object Storage System

ZHAN Ling¹ ZHANG Qiang-shan^{2,3} WAN Ji-guang¹

(Key Laboratory of Data Storage System of Ministry of Education, Wuhan National Laboratory for Optoelectronics, Huazhong University of Science and Technology, Wuhan 430074, China)¹

(School of Computer, Wuhan University, Wuhan 430072, China)² (Xinyang Vocational and Technical College, Xinyang 464000, China)³

Abstract Based on a thorough analysis on the fault-tolerance capability on various existing storage systems, we proposed a new hierarchical, highly reliable, multi-disk fault-tolerant storage system architecture: High Availability Object Storage System (HAOSS). The HAOSS is composed of two layers: the upper-layer and the lower-layer. The upper-layer achieves the high availability by storing multiple replicas for each storage object in a set of storage devices. The individual replicas can service the I/O requests in parallel so as to obtain high performance. But the effective disk space utilization rate for the upper-layer is relatively low. The lower-layer deploys RAID5, RAID6 or RAID_Blaum coding schemes to tolerate multi-disk failures. The disk utilization rate of coding schemes is higher than that of multiple replicas. These advantages come at the price of more complicated fault-tolerant coding schemes, which involve a large amount of calculation for encoding and cause an adverse impact on the I/O performance. The HAOSS puts new objects and hot objects in its upper-layer, so that the majority of the requests are absorbed by the upper-layer, hence achieving guaranteed system I/O performance. The main purpose of the lower-layer is to provide a reservoir for the cold data. In a 1000Mb Ethernet interconnection environment, with a request block size of 1024kB, the sequential read performance for a HAOSS server reaches 104MB/s, which is very close to the theoretical maximum effective bandwidth of Ethernet networks.

Keywords Data organization, High availability, Object storage system, Fault-tolerant

1 前言

近年来在提高存储系统性能方面的研究取得了较大的进展。但是,关于存储系统可靠性的研究更加急迫,因为部分磁盘的失效可能导致整个存储系统上的数据不可用,这是一种

灾难性的后果。最常用的办法,就是通过增加冗余来提高存储系统的可靠性,RAID 磁盘阵列就是一种广泛接受的冗余技术。在 RAID 结构中,通过把用户数据进行编码产生冗余(校验)信息并和用户数据一起保存在磁盘阵列上,以便在故障发生后进行数据恢复。但是随着各类信息爆炸式的增长,

到稿日期:2008-12-25 返修日期:2009-05-15 本文受国家自然科学基金重点项目(编号:60933002)和国家 863 计划重大项目(编号:2009AA01A402)资助。

詹玲(1973-),女,博士研究生,讲师,主要研究领域为网络存储系统;张强善(1974-),男,硕士,讲师,主要研究领域为计算机网络技术、数据库;万继光(1972-),男,博士,讲师,主要研究领域为网络存储系统,E-mail:jgwan@mail.hust.edu.cn。

导致磁盘容量不断扩大,所组建的磁盘阵列的规模也越来越庞大,而随着成员磁盘数目的增加,RAID 磁盘阵列发生磁盘故障的可能性也会增加。许多研究表明,如果 RAID 系统中一个成员磁盘发生故障,那么在未来很短的时间内其它成员磁盘发生故障的可能性非常高。因此,进一步提高磁盘阵列的容错能力成为一个研究的热点。

通过对现有不同存储系统的容错能力进行认真分析,提出了一种新的高可用对象存储系统架构 High Availability Object Storage System (HAOSS)。本文第 2 节介绍了国内外研究状况;第 3 节分析了基于 RAID0_X 的高可用对象数据组织;第 4 节详细描述了 HAOSS 系统的数据组织设计;第 5 节给出了 HAOSS 两层数据组织的设计思路;第 6 节给出了 HAOSS 系统的原型系统搭建;第 7 节对 HAOSS 系统进行了详细的性能测试和分析;最后是结束语。

2 国内外研究状况

2.1 对象存储技术的发展

现有的存储系统一般分为文件存储和数据块存储,文件存储提供给用户一个更高层的逻辑接口,方便不同操作系统平台之间的数据共享,但是文件系统的开销比较大,当存储设备比较多时,文件系统性能受限。数据块存储提供更加快速的数据访问,但是访问层次比较低,共享不方便,需要依赖于元数据服务器的管理。对象存储是文件存储和数据块存储的结合,它将二者的优点有机的结合起来,既能够提供很好的性能,也可以降低元数据服务器的负载。对象存储起源于卡内基梅隆大学 1995 年开始研究的附网安全盘(NASD)项目,2004 年成为 ANSI 标准,名称为“SCSI Object-Based Storage Device Commands (OSD)”。对象具有类似面向对象的访问接口和用户可访问的对象属性,这些属性可以通过对象接口来访问。对象与传统的数据块不同,传统的数据块长度一般是固定,而对象是可变长的。对象存储跟传统的文件类似,而且它的功能更加强大,可以存储任何类型的数据,比如一个文件、文件的一部分、多个文件、一个目录或者数据库的一个记录。另外,对象有属性,数据块是没有属性的,对象的属性要比现有的文件属性更为丰富,而且可以通过自定义的方式进行扩展,方便对不同的文件系统提供支持。

2.2 存储系统的容错技术的发展

目前,国内外对存储系统的容错主要有两种研究思路。

第一种思路:构建组合嵌套 RAID。这种思路是受到了 RAID10 的启发,通过将磁盘阵列组合来达到容许多个磁盘同时故障的情况。因此,出现了 RAID30, RAID50, RAID53 等等组合,经过长期的研究,这些组合在一些特定应用和产品中得到实现,例如早前韩国 Diskband 公司就推出了 TLR(2 层 RAID)的技术,TLR 技术通过把原来单级结构 RAID 改成 2 级 RAID,先用 IDE RAID 芯片接上 2 个普通 IDE HDD,实现 RAID1。再把每个 IDE RAID1 作为一块硬盘,再次实现了 RAID0,1,0+1,3,5,这种构造在读写速率、可靠性等方面有了比较大的改善,类似这种组合嵌套的 RAID 技术现在还在不断发展,而多层组合 RAID 则是这个方向上的进一步深入。

此种方案的优点是提供了很好的顺序和随机读写能力,由于能够同时对多个磁盘进行读写,这些组合嵌套 RAID 通常会比同等的单级 RAID 的性能更加优秀,同时能允许两个

或者两个以上的磁盘故障;缺点是这些阵列组的实现比较复杂,实施起来费用也比较高。一个磁盘故障后的重建工作可能会严重影响整个阵列的性能,而且这些阵列所谓的允许两个或两个以上的磁盘故障是在“不同容错组内”这个大前提下的。

第二种思路:提出新的能够承受两个甚至两个以上的磁盘故障的 RAID 结构。伴随着这种思路,一系列针对 Double Parity 的编码算法应运而生,主要有 RM2, RS, EVENODD, RDP 等,越来越多的磁盘阵列厂商开始推出“允许两块磁盘损坏”的 RAID 产品,如 NetApp 的 NearStorage, HP 的 ProLiant 服务器和 MSA 系列以及 IBM 的 RAID5E, RAID5EE 等。RAID6 也在此种情况下被提出,这种 RAID 级别是在 RAID 5 的基础上发展而成,因此它的工作模式与 RAID 5 有异曲同工之妙,不同的是 RAID 5 将校验码以左旋的方式写入到各驱动器里面,实际数据盘是阵列盘数-1,而 RAID 6 将校验码写入到两个驱动器里面,这样就增强了磁盘的容错能力,同时 RAID 6 阵列中允许出现故障的磁盘也就达到了两个,但相应的阵列磁盘数量最少也要 4 个,目前很多大型企业都开始实现 RAID6 这一新的 RAID 级别。

而能够承受多个磁盘故障的 RAID 结构仍在继续探索中,这部分探索主要是对于新的编码方式的研究,如 Blaum 码、Hover 码、EEDD 码等,应用这些编码方式的 RAID 结构都能够从理论上解决 3 个甚至以上的磁盘故障恢复问题,但是都不同程度上存在一些弊端: Blaum 码的容灾能力随着冗余磁盘数的增加而增加,但这种结构的主要问题在于解码方法是解多项式环上的一组线性方程,解码算法不易实现,并且解码复杂度高;Hover 码也是一种能够承受多个磁盘故障的 RAID 结构,这种结构的主要问题是冗余盘数目并不是随着磁盘阵列系统总盘数线性增长,冗余信息量太大、代价昂贵。由于编码解码复杂度大大提高,对阵列读写性能,尤其是写性能影响比较大,而且当有磁盘损坏时,恢复的周期长。这些问题的存在导致这种思路还没有对应的成熟产品问世。

3 基于 RAID0_X 的高可用对象数据组织

通过之前的分析,要构建一个高可用对象存储系统,首先要解决的最主要的问题就是读写性能问题,在已有的存储系统中,RAID10 是已经很成熟的一种存储架构,是存储性能和数据安全兼顾的方案。可以简单概括为: RAID0 是提速, RAID1 是备份。这样的组合既可提高存储系统的速度性能,又可提供数据安全性,它在提供与 RAID1 一样的数据安全保障的同时,也提供了与 RAID0 近似的存储性能。

现在很多分布式并行文件系统都采用 RAID10 的数据组织方式。虽然 RAID10 的存储性能较好,但是在磁盘容错方面只能容许一个盘出错,还远没有达到多磁盘容错的要求。因此,提出一种新的 RAID0_X 的数据组织结构,如图 1 所示。阵列 RAID0_X 由 X 个 RAID0 组成,其中后 X-1 个阵列都是第一个 RAID0 阵列的备份,但是并不是全部数据的备份,而是根据用户设置的数据的安全级别来决定。用户可以存储到 RAID0_X 中数据设置 X 种安全级别,数据的安全级别每提升一级,该数据在 RAID0_X 阵列中的备份增加一份。可见,X 的大小由具体需求的最高可靠性级别决定,而且 RAID0 和 RAID10 都是 RAID0_X 的特例,RAID0 对应

RAID0_1, RAID1 对应 RAID0_2。

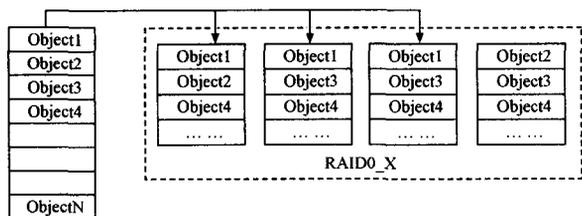


图1 RAID0_X的数据组织结构

RAID0_X与RAID10不同之处是,它不是一个真正意义上的RAID,数据组织以对象为单位,每个对象可以有多个备份,比如Object1,有2个副本, Object2有1个副本, Object4有3个副本。这样通过对象的副本数属性来控制每个对象的副本数,从而不同的对象可以有不同的磁盘容错能力。采用并行写方式,RAID0_X的写性能和RAID0相同,而由于有些数据有多个备份,可以并行从这多个备份中读取,因而读性能会有很大的提高。

4 HAOSS系统的数据组织设计

RAID0_X虽然具有非常好的读写性能,而且具有灵活的数据安全级别,但是它的缺点也十分明显,数据存在多个副本,磁盘空间利用率只有1/X。而采用编码方式,写性能受限,但是,磁盘利用率比较好。结合以上两种思路的优点,研究出一种既包含两者的优点,又能最大限度地克服两者缺点的高性能磁盘容错结构。通过分析不难发现,要想使以上两种思路能够结合起来,要解决的最重要的问题就是如何保证整个存储系统的读写性能。在这里,HP的AutoRAID存储方案给了我们一个很好的启示。

AutoRAID具有一个重要的特点——动态数据迁移。AutoRAID技术懂得RAID级别的差异并动态适应主机的需求来优化性能。磁盘控制器能转移存储在一个RAID级别的数据到另一RAID级的数据存储区中,并保持对主机透明。这种动态的数据迁移策略使常用数据保存在RAID10(提供最好总体性能),不常用数据放在RAID5中(低费用)。通常在一定的时间内常用数据只占一小部分,所以这种策略使AutoRAID具有RAID10的性能、RAID5的费用。

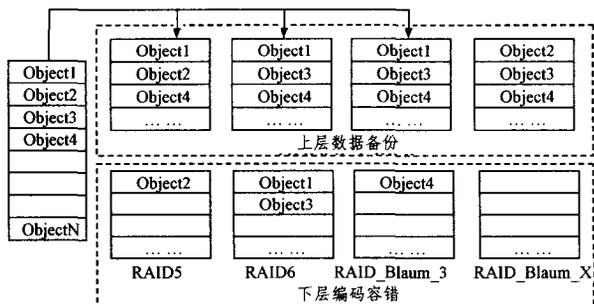


图2 HAOSS数据组织结构

通过对AutoRAID的特点以及现有的各种磁盘阵列结构的分析,我们将提出一种更加灵活可靠的高可用磁盘容错存储系统HAOSS架构。HAOSS架构分两层:上层是通过RAID0_X的多个数据备份来达到多盘容错功能,下层是通过采用不同的容错编码来实现多盘容错功能。在同样的容错级别下,上层有很高的性能,但是磁盘利用率低;下层的磁盘利

用率高,但是性能低。采用这两级方式,将常用的数据放上层,不常用的数据放下层,即保证了系统的性能比较高,并且具有较高的磁盘空间利用率。HAOSS数据组织结构如图2所示。

HAOSS上层采用RAID0_X结构,优点是具有和RAID0相同的写性能,读性能比RAID0更高,并且最多能容许X-1个磁盘同时发生故障,而缺点是磁盘空间利用率低。下层由RAID5,RAID6以及RAID_Blaum组成(这里的RAID_Blaum定义为通过Blaum码实现的RAID阵列结构,Blaum码的容灾能力随着冗余磁盘数的增加而增加,即当有3个冗余盘的时候,RAID_Blaum能满足3个磁盘同时故障时的数据恢复(即图中的RAID_Blaum_3),而如果有X个冗余盘的时候,RAID_Blaum的容错能力就为X个盘(即图中的RAID_Blaum_X),它的磁盘空间利用率为(N-X)/N,N越大,空间利用率越高,因此适用于大规模的磁盘阵列系统。下层的特点为:优点是容错能力强,RAID5能容错一个磁盘的故障,RAID6能容错两个磁盘的故障,RAID_Blaum容错能力随着冗余盘的数目增加而增加,同时它们的磁盘利用率较高,而且随着阵列组的规模增大而提高;缺点是由于编码越来越复杂,性能受到极大的影响。

5 HAOSS两层数据组织的设计思路

(1)容量分布:上面的RAID0_X部分占用整个磁盘阵列组总磁盘空间的10%~20%或者更低(由具体的阵列规模决定),而下面的RAID5,RAID6和RAID_Blaum部分则占用整个磁盘阵列组总磁盘空间的80%~90%或者更高。这样,磁盘阵列组的磁盘空间利用率主要由下面部分决定,因为下面部分的磁盘利用率高,所以整体的磁盘利用率也会处在一个较高的水平。

(2)写操作:所有用户数据在写入该阵列组的时候,都是首先写入到上面部分的RAID0_X阵列中,并且根据用户对不同数据设定的安全级别,决定该数据写入到RAID0_X中的备份数目(注意:对于多个数据备份,系统会自动将各备份写到适合的RAID0阵列中,并不一定是按照顺序放置的)。如此可以对可靠性不同数据进行不同的安全保护,采用并行写技术,写入数据的性能和RAID0一样高。

(3)数据迁移:由于所有写操作都是在RAID0_X中进行的,而整个阵列组保存数据的空间主要是在下面部分,因此必须进行上下两部分的数据迁移。迁移的规则由上下两部分相同的安全级别之间进行数据迁移,如图2所示,安全级别为2的数据一定有二份分别存储在不同的RAID0中,所以这个安全级别的数据的迁移是将这两个RAID0中的任何一个迁移到下面部分的RAID5(这些数据的安全特性没有改变,仍然是只允许一个磁盘故障),迁移完成后将这两个RAID0中对应的那份数据删除;而所有安全级别为3的数据,则是将3个RAID0中的某一数据迁移到下面部分的RAID6中,同时删除3个RAID0中对应的数据备份,以此类推。这样,可以保证数据在进行迁移之后安全性能没有改变。数据迁移的时机:当RAID0_X中的某部分数据热度较低的时候则对这部分数据进行迁移(数据热度是指某部分数据在一段时间内的访问频率,访问频率高,则热度也高),并且是在系统相对空闲时进行。

(4)读操作:当用户进行读操作时,首先从上面部分的 RAID0_X 中查找数据,如果命中,则读性能比 RAID0 还要高;如果没有命中,则再从下面部分进行查找,此时读性能较低。但是由于 RAID0_X 中保存的是热度相对较高的数据,因此读数据时的命中率会比较高,因而整个存储系统将有较高的读性能。

综上所述,对该存储系统特点总结如下:

(1)适用于数据量较大的大型存储应用。

(2)容错能力极强,理论上可以满足任意多个磁盘故障时数据的恢复,同时用户可以对数据设置不同的安全级别,具有极大的灵活性。

(3)读写性能优秀。写性能和 RAID0 相同,读性能随着 RAID0_X 命中率的提高而提高,命中时的读性能比 RAID0 要高很多。

(4)具有较高的磁盘空间利用率,因为阵列组的主要容量分布在磁盘利用率很高的 RAID5,RAID6 部分,所以整体的磁盘空间利用率较高。

6 HAOSS 系统的原型系统

为了验证上述高可用对象存储系统的数据组织策略,我们实现了一个 HAOSS 原型系统。原型系统的整体结构如图 3 所示。整个系统可以分为应用服务器和存储服务器两大部分。应用程序的 I/O 请求首先送给文件系统,文件系统把对文件的访问转换为对 OSD 对象的访问,发给 OSD 启动器。OSD 启动器收到请求之后,再转换为标准 SCSI 命令,通过 iSCSI 启动器和 TCP/IP 转发给存储服务器。而 OSD 目标器负责处理 iSCSI 目标器接收的 SCSI 命令。当是数据读写命令时,首先查询 Cache,如果命中,就直接返回请求的数据;如果不命中,就将请求发给 RAID0_X 模块,在 RAID0_X 模块中查询需要的数据,如果命中,就返回需要的数据;如果不命中,再从容错编码模块中查询需要的数据,并把查询到的数据迁移到 RAID0_X 部分。

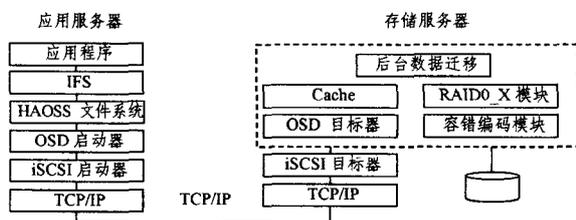


图 3 HAOSS 原型系统结构

后台数据迁移模块负责在 RAID0_X 存储池和容错编码存储池之间迁移数据。当一个 RAID0_X 存储池中的数据长时间不使用,就可以把它迁移到容错存储池中,可以节省一定的存储空间。同时,当容错存储池中的一个数据又被多次访问,就把它迁移到 RAID0_X 存储池,以提高访问性能。

7 性能测试与结果分析

7.1 测试平台

为了验证 HAOSS 的性能,在 1000MB 以太网环境下,分别对不同的请求进行了测试。测试是在 2 台通过 1000Mbps 的 IP 交换机连接起来的 PC 机上进行的,其中 1 台是 HAOSS 服务器,另外 1 台是客户机。2 台计算机的配置如表 1

所列。

表 1 测试环境配置

操作系统	测试服务器:Windows Server 2003 SP1。OSD 服务器为 Fedora Core 8.0
Cisco 3750 交换机	24 个千兆以太网铜缆接口
硬盘型号	Maxtor STM3320820AS 硬盘 6 个,容量 320GB,转速 7200RPM
主板	SUPER X7DVL-I
CPU	INTEL 双核至强 5110 1.6GHz
网卡	Intel 1000M 网卡
内存	1G DDR2
iometer	2003.05.10,测试时间为 1 分钟

7.2 性能测试及分析

图 4 和图 5 为 SMB 和 HAOSS 顺序读写性能的测试结果,从中可以看出 HAOSS 的顺序读写性能都要高于 SMB。当请求比较小时,HAOSS 性能比 SMB 高出不多,但是随着请求数据块的增加,HAOSS 性能明显高于 SMB,当请求达到 512kB 时,HAOSS 顺序读写性能要比 SMB 高 50%。在 1000MB 的以太网环境下,当请求块大小达到 1024kB 时,HAOSS 的顺序读写性能达到 104MB/s,达到 1000MB 以太网的理论最大带宽。从而说明 HAOSS 提供多盘容错情况下,能够提供一个很好的性能。这主要是因为顺序读写的环境下,读写请求主要在上层命中,上层本身的性能比较好,从而存储系统的整体性能就比较好。

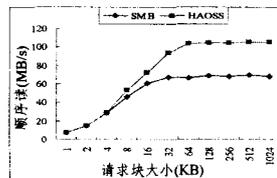


图 4 顺序读性能测试结果

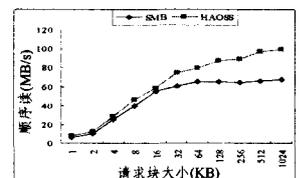


图 5 顺序写性能测试结果

图 6 和图 7 分别为随机读和随机写性能测试结果,从中可以看出 HAOSS 的随机读和随机写性能都比 SMB 低一些。这是因为随机读写情况下,HAOSS 在上层的命中率比较低,很多请求需要访问下层,从而对性能有一定的影响。另外,数据迁移模块会根据请求来从下层预期一定的数据到上层,而随机访问情况下,这些预期命中的可能性很小,从而也影响了一定的性能。

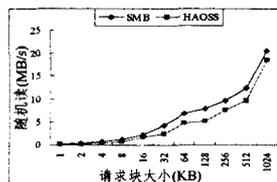


图 6 随机读性能测试结果

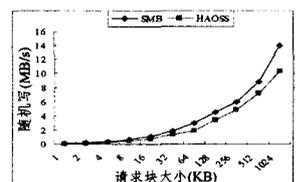


图 7 随机写性能测试结果

结束语 综上所述,HAOSS 系统针对性强,可支持多盘容错的对象存储系统。该系统采用分层存储结构,上层采用多个备份的方式保证性能和容错级别,下层采用容错编码实现容错级别和高磁盘利用率。在数据组织上,新对象和热点对象都是放上层,这样大部分请求都能够在上层命中,从而保证系统的性能。下层主要用来存放不常用的数据。采用这种分层的结构,保证了系统的性能比较高,并且具有较高

(下转第 130 页)

组通过事件端口控制进程状态之间的迁移。在图中将错误的状态和状态间的转移用灰色线表示以便区别。

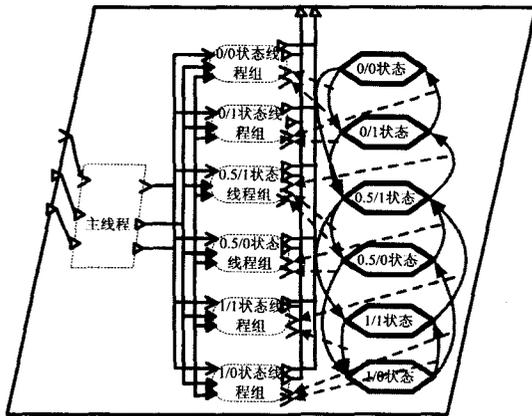


图4 智能窗帘控制系统AADL模型

根据AADL模型转换的无概率马尔可夫链模型如图5所示。

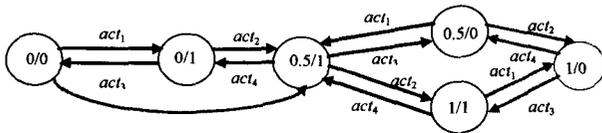


图5 智能窗帘系统无概率马尔可夫链模型

根据图3的期望马尔可夫链,依概率产生50组输入序列,这些输入序列组成了测试用例,并结合以往测试经验设定该被测模型的 $D(U, T)$ 的阈值 $d=0.03$,根据上述算法计算出实际马尔可夫链模型信息表如表2所列。

表2 实际马尔可夫链模型信息表

起始状态	事件编号	事件描述	目标状态	转移概率
S ₀	e _{0,1}	act ₁	S ₁	0.8
S ₁	e _{1,2}	act ₂	S ₂	0.65
S ₁	e _{1,0}	act ₃	S ₀	0.35
S ₂	e _{2,3}	act ₃	S ₃	0.25
S ₂	e _{2,4}	act ₂	S ₄	0.4
S ₂	e _{2,1}	act ₄	S ₁	0.35
S ₃	e _{3,2}	act ₁	S ₂	1
S ₄	e _{4,2}	act ₄	S ₂	1

(上接第126页)

的磁盘空间利用率。两层之间的动态数据迁移是下一步研究的重点。

参考文献

- [1] Patterson D, Katz R. A case for redundant arrays of inexpensive disks (RAID). Chicago, 1988; 109-116
- [2] Blaum M. EVENODD: An optimal scheme for tolerating double disk failures in RAID architectures[J]. IEEE Transactions on Computers, 1995, 44(2): 245-254
- [3] Blaum M. New array codes for multiple phased burst correction[J]. IEEE Transactions on Information Theory, 1993, 39(01): 66-77
- [4] Hafner J L. Hover erasure codes for disk arrays[R]. RJ10352 (A0507-015). IBM Research Division, July 2005
- [5] Hafner J L. WEAVER codes: Highly fault tolerant erasure codes for storage systems[C]// Proceedings of the FAST-

结合表1,表2使用公式 $D(U, T) = \sum_{ij} \pi_i p_{ij} \log_2 \frac{p_{ij}}{p_{ij}}$, 计算

分析实际模型与期望模型之间的差异,得到 $D(U, T) = 0.0652$. 设定的阈值 $d=0.03$, $D(U, T) > d$, 说明实际模型与期望模型的差异较大, 实际马尔可夫链模型中存在错误, 即由此映射的AADL模式转移有错误。

结束语 本文首先介绍了基于模型的测试的基本概念和基本原理, 进而提出了一种结合马尔可夫链模型对AADL模型进行测试的方法, 并给出示例对这种方法进行了说明。由于AADL模型没有对状态迁移概率进行描述, 因此应用本文的测试方法, 能够更加精确地从数学统计角度对测试结果进行分析。由于对基于模型的测试的研究尚处于早期阶段, 因此对AADL模型进行测试还有很多问题需要解决, 比如测试用例的生成及优化, 测试的覆盖率问题, 测试自动化方法的研究以及测试结果的分析评估等, 这些都需要进行进一步的研究。

参考文献

- [1] OMG. Model Driven Architecture(MDA)[OL]. <http://www.omg.org/mda/>, 2002
- [2] Feiler P H, Gluch D P, Hudak J J. The Architecture Analysis & Design Language(AADL): An Introduction[M]. Carnegie Mellon University, 2006
- [3] Pretschner A. Model Based Testing[C]// ICSE'05. St. Louis, Missouri, USA, May 2005
- [4] Conformiq Software Ltd. Conformiq Test Generator[OL]. <http://www.conformiq.com/products.php>
- [5] Reactive Systems Inc. Reactis[OL]. <http://www.reactive-systems.com/>
- [6] MIT Software Design Group. MulSaw[OL]. <http://projects.csail.mit.edu/mulsaw/>
- [7] University of Twente, Formal Methods and Tools research group. TorX[OL]. <http://fmt.cs.utwente.nl/tools/torx/introduction.html>
- [8] 陈祎. 基于UML模型的软件测试技术研究实现[M]. 成都: 电子科技大学, 2006
- [9] Whittaker J A, Thomason M G. Senior - Member -, IEEE A Markov Chain Model for Statistical Software Testing[J]. IEEE Transactions on Software Engineering, 1994, 20(10)

2005; 4th Usenix Conference on File and Storage Technologies. December 2005

- [6] Bloemer J M, Kalfane M, Karpinski R. An XOR-based erasure-resilient coding scheme[R]. ICSI TR-95-048
- [7] Alvarez G A, Cristian F. Tolerating multiple failures in RAID architectures with optimal storage and uniform declustering. Colorado, 1997; 62-72
- [8] Jin Fan. An investigation on new complex rotary codes. Brighton, UK, 1985; 1-8
- [9] Feng Dan, Zhang Jiang-ling. Improved EVENODD code. Ulm, Germany, 1997; 261-262
- [10] Dong Huan-qing, Li Zhan-huai. RAID-VCR: A New RAID Architecture for Tolerating Triple Disk Failures[D]. School of Computer Science, Northwestern Polytechnical University, Xi'an
- [11] Wikes J, Golding R, Staelin C, et al. The HP AutoRAID Hierarchical Storage System. Hewlett-Packard Laboratories