第33卷 第16期 计 算 机 工 程 2007年8月
Vol.33 No.16 Computer Engineering August 2007
·开发研究与设计技术· 文章编号：1000—3428(2007)16—0269—03 文献标识码：A 中图分类号：N945

基于Linux的RAID在线扩展功能的设计与实现

齐 路，谢广军，刘晓光，王 刚，刘 璟
（南开大学计算机系，天津 300071）

摘 要：基于Linux设计并实现了独立冗余磁盘阵列(RAID)的在线扩展功能，保证了对计算机存储系统进行扩展时用户请求的不间断访问，提高了系统的可用性。讨论了在利用普通方法对软件RAID进行结构扩展的基础上实现在线扩展功能，以及扩展过程中数据迁移和用户请求间的冲突问题的解决策略。试验表明，在线扩展操作在多次运行中稳定可靠，性能也较为理想。

关键词：Linux；软件RAID；在线扩展

Design and Implementation on RAID On-line Expansion Based on Linux

QI Lu, XIE Guang-jun, LIU Xiao-guang, WANG Gang, LIU Jing
(Department of Computer, Nankai University, Tianjin 300071)

【Abstract】 This paper designs and implements on-line expansion function in RAID system base on Linux, which allows RAID system to keep receiving requests from user while expanding. This function promotes the usability of the system. It also describes the implementation of on-line characteristic while using the simple method for RAID structure expansion and the solution to deal with the conflict between data migration and user's requests. Experiments show that on-line expansion is stable in a long time running and the performance is acceptable.

【Key words】 Linux; software RAID; on-line expansion

随着计算机技术和Internet的迅速发展，各种应用需要存储的数据量也在迅速膨胀，当存储系统的容量不能满足用户需求时，就必须对存储系统进行扩展。但对于一些关键应用，例如银行、医院和铁路等部门，必须保证不间断的服务，否则将造成巨大损失。因此，存储系统的在线扩容成为存储系统领域近年的热点。

目前，主流存储设备厂商，例如EMC、HP等推出的可扩展存储系统基本都是基于特殊硬件设备实现的。受硬件设备的限制，这些“可扩展”存储系统一般扩展能力有限，不能实现无限制扩展。与基于硬件的存储系统扩展相比，基于软件的存储系统扩展具有实现灵活、成本低廉、可多次扩展、理论上具有无限扩展能力等特点，能达到网络数据服务对于存储系统的要求。

本文针对应用最为广泛的存储系统数据布局方式——独立冗余磁盘阵列(RAID)[1]，在Linux的软件RAID存储系统基础上，实现了RAID0和RAID5的在线存储扩展功能。本存储系统除了具有配置灵活、可以根据用户需要实现重复扩展等特点以外，它对于上层的文件系统和用户的应用是透明的，上层应用不需要作任何的修改就可以直接使用。

1 RAID扩展涉及的问题

对于RAID5以及能够双容错的RAID6而言，由于它们的校验数据散布于各磁盘间，因此对它们的结构进行扩展时需要对条纹进行重新组织。

扩展后新增磁盘能够分担原阵列磁盘的负载，使得用户请求产生的正常负载或者其他原因产生的额外负载，在整体上都能达到负载均衡。

新增磁盘的空间应当通过某种关系映射到扩展后的阵列中，因此，可以通过改变映射方式减少RAID扩展中所需的大量数据迁移，“新增空间散布的扩展方法”[2]采用的就是类似的方式。但采用这种方式扩展的RAID并非对于所有访问模式都有出色的性能。

2 设计目标和设计思想

2.1 设计目标

RAID在线扩展功能的设计，需要重点考虑4个方面：(1)扩展前后以及扩展过程中应保持数据的一致性，并且不能影响系统服务的正常运行。数据一致性是指扩展的进行与否对系统上层模块访问（例如文件系统）是透明的，提供给上层模块的数据视图不发生变化。(2)在扩展过程中，无论处于什么状态，用户对数据的访问都能够得到及时、正确的响应。(3)为系统管理员提供完备的管理工具，实现包括启动、暂停、回退以及系统异常（例如系统掉电、关机）处理和恢复等功能。(4)服务质量的要求，扩展的总时间和扩展过程中对数据访问的响应时间都应是用户可以接受的。

2.2 设计思想

本文实现的RAID扩展采用的是平凡扩展方法。平凡方法是指扩展后的磁盘阵列保持原阵列的数据布局方式不变，同时移动位置发生变化的数据。这种方案的优点在于：(1)由

基金项目：国家自然科学基金资助项目(90612001)；天津市科技发展计划资金资助重点项目(043800311，043185111-14)；南开大学创新基金资助项目

作者简介：齐 路(1983－)，男，硕士研究生，主研方向：并行与分布式系统；谢广军，博士研究生；刘晓光、王 刚，副教授；刘 璟，教授、博士生导师

收稿日期：2006-09-11 **E-mail**：tidyroad@163.com

于将原阵列的数据分布到所有磁盘，包括新增磁盘，而新增的空闲地址空间也分布到所有磁盘，因此阵列中负载是均衡的；(2)扩展后仍然采用原阵列数据映射方法，实现简单，可用性高；(3)由于扩展后得到的是标准的阵列结构，可以实现连续的扩展。

3 系统设计与实现

3.1 总体结构

整个功能的实现分为 3 个模块：管理模块，数据迁移模块，访问控制模块。管理模块接收用户进行在线扩展的命令和参数，将其通过系统的 IOCTl 接口由用户态转入内核态。在内核态中，管理模块对参数进行一些必要的处理并设置 RAID 系统的状态，然后将剩余的工作交由数据迁移模块继续完成。数据迁移模块是一个内核线程，每次循环都要检测当前 RAID 系统的状态，根据不同状态完成不同工作。访问控制模块用于处理 RAID 扩展过程中上层发来的用户请求。

3.2 系统实现及其关键技术

3.2.1 数据迁移

(1)设置迁移粒度的最小值。若 m、n 分别为新老阵列数据盘的个数，b 为条纹单元大小，则最小迁移粒度 g 为

$$g=[m, n]\times b \tag{1}$$

其中，$[m, n]$代表 m 和 n 的最小公倍数。这样保证了在一次读写拷贝中的数据在原阵列和新阵列中都是整条纹的，有利于 RAID5 结构计算校验，也最大限度地保证了各个磁盘的负载均衡。具体的迁移粒度为 g 的整数倍，由用户命令参数指定，增加了系统的灵活性。

(2)迁移过程中系统管理员能够通过管理模块控制扩展的动作。为此，将迁移模块实现为一个内核线程，需要迁移数据时将其唤醒。同时还设置了 6 个扩展状态，迁移模块每次在对一个迁移单元进行操作前，都将检测扩展状态以决定下一步的操作。各状态间的转换关系如图 1 所示。

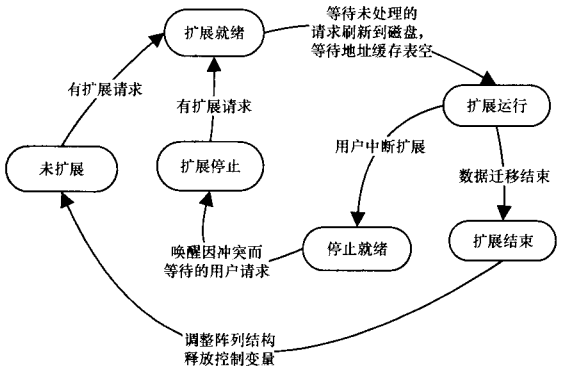

图 1 扩展状态转换图

RAID 的初始状态为“未扩展”，当系统管理员通过管理模块发送扩展启动命令时，RAID 进入“扩展就绪”状态，在等待阵列中当前的 I/O 请求下发至磁盘，同时等待地址缓存表初始化完毕，这样避免了扩展开始就与用户请求发生冲突。一切准备就绪，将系统置为“扩展运行”状态，开始数据的迁移。全部数据迁移完毕后，系统进入“扩展结束”状态，此时将系统中阵列控制结构相关字段置为扩容之后的值，系统恢复到初始状态。

3.2.2 用户请求和数据迁移的协调

对于数据迁移过程中用户 I/O 请求的处理，可以分解为 3 部分，如图 2 所示。若用户请求位于区域 1，则待访问区域已经进行了结构调整，应按照扩展后阵列的地址映射函数对其地址重新映射；若用户请求位于区域 3，则待访问区域还没有进行结构调整，用户请求的地址应按扩展前阵列的地址映射函数进行映射；若用户请求位于区域 2，则需要等待当前的迁移区域迁移完毕后再进行处理。

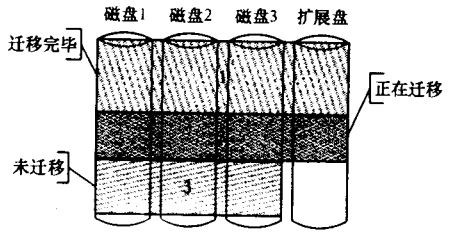

图 2 用户请求处理

在扩展的过程中，访问控制模块和数据迁移模块都需要检测用户请求和数据迁移是否发生冲突，即用户请求的地址是否与当前的迁移区域交叠。因此，访问控制模块须比较当前的迁移位置与当前的用户请求的地址，若有冲突则设置请求标志后睡眠，等待当前迁移操作完成。迁移线程完成一个迁移单元后，检测该迁移单元存在因冲突而阻塞的读写请求，便唤醒用户请求。对于迁移模块，冲突检测是一个一对多查找操作，为此，内存中维护着一个地址缓存表，用于记录可能发生冲突的地址，同时运用开散列的方法以提高查找效率。地址缓存结构中记录了迁移区域的起始地址和在此区域上用户请求的个数。在每次数据迁移操作的开始，搜索地址缓存表，若存在访问冲突则睡眠等待用户请求的完成。用户请求完成后，在回调函数中将尝试唤醒睡眠的迁移线程。迁移线程递减地址缓存结构中的引用计数，直至为 0 后将其删除。

3.3 中断与恢复

扩展过程中，允许管理员中断扩展过程。如图 1 所示，系统先进入“停止就绪”状态，唤醒因冲突而等待的用户请求后进入“扩展停止”状态。此时，若用户请求与迁移区域相交，迁移过程的停止将导致用户请求区域的前一部分已经完成数据迁移，而另一部分还没完成。对这种情况，需要将用户请求拆分成 2 部分分别处理。

在每次成功迁移一个迁移单元后，迁移模块将把扩展信息写入超级块，并将超级块写入磁盘。这使得在扩展过程突然掉电的情况下，重启后能继续完成扩展过程。但在最初的几个迁移单元中，若突然掉电可能会造成数据被覆盖、扩展不能重新开始。虽然这种情况发生的概率较小，但本文还是提供给用户一种解决方案：在最初的迁移过程中，先对迁移的数据进行备份。设原阵列数据盘数为 m，校验盘数为 n，新添加的磁盘数为 k，一个迁移单元中条纹单元数为 i，x 为需要备份的迁移单元数。则应满足：

$$\frac{x\cdot i}{m-n}-\frac{x\cdot i}{m-n+k}\geq\frac{i}{m-n+k} \tag{2}$$

$$x=\left\lceil\frac{m-n}{k}\right\rceil \tag{3}$$

4 实验结果与性能分析

实验对 RAID 在线扩展功能的正确性和扩展过程中 RAID 对I/O请求的读写性能进行了测试，测试主机为 CPU P4 2.0GHz，内存 256MB，操作系统为 Redhat 企业版 4.0。

先在一个硬盘上划分 5 个大小为 300MB 的分区作为模拟的 RAID 磁盘，在其中 3 个 RAID 盘上建立 RAID5 系统，另 2 个作为扩展盘。利用笔者自行开发的 Exponline 测试程序向

RAID 和系统中的某个分区中同时写入数据并在这个范围内随机地进行一些读写操作。根据扩展、停止、重启、回滚等事件发生的位置和时间的不同所进行了 30 余种情况测试结果表明，RAID 扩展后，数据正确无误，即使长时间运行也稳定可靠。

经测试在完成时间方面，在没有用户请求的情况下，在线扩展的完成时间约为降级模式下同步操作的 2 倍；在读写性能方面，利用 Iozone 测试软件测试了扩展模式和同步模式下 RAID5 对用户 I/O 请求的读写性能，如图 3 所示。可以看出，对于 64KB～1MB 不同大小的用户请求，在 2 种模式下用户请求的写速率几乎相同，而扩展模式下的读速率约为同步模式的一半。所以在性能方面，应该在用户可以接受的范围内。

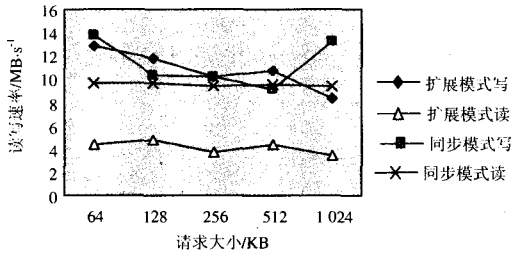

图 3　用户读写请求性能比较

5　结论和改进方向

本文所述的 RAID 在线扩展功能，保证了 RAID 系统在容量扩展的同时用户请求的不间断访问，同时扩展过程保持了原有的数据布局方式，系统能够进行连续的扩展，且扩展过程中与扩展结束后都能达到较好的负载均衡，提高了系统的可用性。通过测试表明，本文所述的实现方式不仅能够正确地完成数据的扩展，而且能够较好地协调数据迁移和用户请求。在与 Linux 自身的软 RAID 驱动程序的同步操作的比较中，二者对于用户的写请求性能无明显差异，但读性能有待进一步提高，同时，扩展实现中对于磁盘损坏等容灾处理还不够完善，这些都是今后的改进方向。作为一个重要的功能模块，本文所实现的 RAID 在线扩展系统已经成为了台湾英业达集团所推出的存储系统卷管理器的一部分，并且这一产品已经通过了相关测试，将在近期推向市场。

参考文献

1 Patterson D A, Gibson G. A Case for Redundant Arrays of Inexpensive Disks(RAID)[C]//Proc. of 1988 ACM SIGMOD Int'l Conf. on Management of Data, New York. 1988.

2 王　刚, 刘晓光, 刘　璟, 等. 一种新的 RAID 结构快速扩展方法[J]. 计算机工程与应用, 2002, 38(4): 14-16.

3 王　刚. 网络磁盘磁盘阵列结构和数据布局研究[D]. 天津: 南开大学计算机系, 2002.

4 Love R. Linux Kernel Development[M]. 2nd ed. USA: Novell Press/ Pearson Education, 2005.

5 Corbet J, Hartman G K, Rubini A. Linux Device Drivers[M]. 3rd ed. USA: O'Reilly, 2005.

(上接第 268 页)

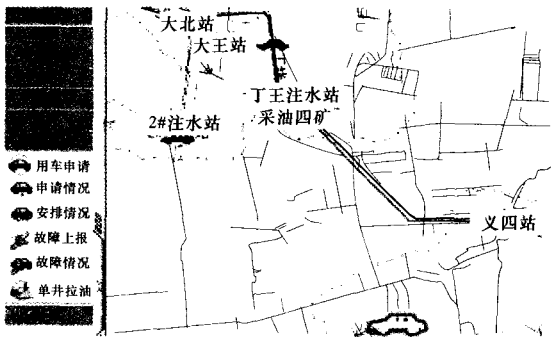

图 3　车辆安排分布

(3)故障影响分析

油田的生产设施分为油、气、水、电、路、讯等多种类型，某个设施发生故障，会影响与其连通或控制的其他设施，因此，用户要快速了解该故障对整个生产系统的影响程度，及时采取措施。生产设施对整个生产系统的影响取决于它在本专业系统控制逻辑中的地位及各专业系统之间的控制关系，这种关系二叉树形式存放在属性库中。当某个设施发生故障，在 Web Server 上的搜索功能按照这种关系找到所有受影响的设施，搜索到的要素就是需要在临时图层上表现的图形要素，由此建立空间请求，即用 WebGIS Service 空间描述语言描述临时图层显示的要素、标注及各类型要素的图符表示、背景数据层(如地形地貌)等，提交 WebGIS Service。WebGIS Service 创建临时图层后，在临时图层中用指定的图符、颜色绘制要素，并进行文字标注，叠加背景层等，形成故障影响范围区域图，辅助故障分析。

空间查询、空间量算、缓冲区分析也可用类似方法实现。由于多个临时图层可以同时存在，表达复杂的组合分析应用时，每种分析结果可以存放在不同的临时图层中，实现不同分析结果的叠加，也可以将合成的结果放在一个临时图层中，表达综合效果。

3　结束语

尽管 B/S 模式下的 WebGIS 技术和应用仍不成熟，但利用临时图层技术在一定程度上能够扩展它的实用功能，使 WebGIS 不再仅局限于简单的图形显示和查询，在数据采集、动态标记、复杂空间分析、业务表达等方面也能发挥较大的作用。实践证明详细了解业务，寻求 GIS 技术和实际应用的契合点，恰当地运用临时图层技术，才能更好地发挥 B/S 模式下 WebGIS 的作用。

参考文献

1 邬　伦, 刘　瑜, 张　晶, 等. 地理信息系统——原理、方法和应用[M]. 北京: 科学出版社, 2003-06.

2 一种基于 B/S 结构与 C/S 结构结合的新体系结构[Z]. (2005-04-10). http://lunwen.zhupao.com.

3　ESRI Inc.. ArcXML Programmer's Reference Guide[Z]. 2002.

4　ESRI Inc.. Customizing ArcIMS——Java Viewer[Z]. 2002.

5　ESRI Inc.. Customizing ArcIMS——HTML Viewer[Z]. 2002.

6 MapInfo Inc.. MapInfo MapXtreme for Java 4.5 产品简介[Z]. (2006-01-31). http://www.gissky.net.