基于块I/O的RAID设计

万亚平,欧阳利军,肖建田,刘 立
(南华大学 计算机科学与技术学院,湖南 衡阳 421001)

摘 要:磁盘阵列(RAID)是当前能够提供存储系统高可用性和高可靠性的一项重要技术。它通过软硬件的冗余和奇偶校验提供数据的重构和恢复。针对当前在RAID控制软件设计的过程中面临多次数据拷贝的问题,文中提出了一种基于块I/O的RAID系统设计。它利用最新的Linux内核所提供的BIO机制,插入到SCSI Target的中间层进行数据I/O的处理。它能屏蔽掉上层不同的设备驱动类型,提供到IP-SAN的无缝链接。实验表明,该设计能够减少数据的传输延迟,最大限度地提高数据传输过程中的吞吐率,避免了多次昂贵的内存拷贝操作。

关键词:磁盘阵列;块设备;吞吐量;响应时间
中图分类号:TP311 **文献标识码**:A **文章编号**:1673-629X(2008)03-0135-04

Design on RAID Based on Block I/O

WAN Ya-ping, OUYANG Li-jun, XIAO Jian-tian, LIU Li
(School of Computer Science and Technology, Nanhua University, Hengyang 421001, China)

Abstract: RAID is an important technology which provides high availability and high reliability storage system. It can provide data reconstruction and restoration by software and hardware redundancy or parity. In this paper, according to the question of the designing of RAID system faced, the solution presented a RAID design which is based on block I/O. It helps to distinguish all kinds of requests and to analyze all kinds of parameters by inserting into the dev handler of SCSI target level and will be able to achieve a seamless IP-SAN Links. Through direct block I/O, it can reduce a memory copy. From the results, shows that the design can decrease data transfers delay and farthest improve data throughout in the process of transfer. This also can avoid repetitious costly memory copy operation.

Key words: RAID; block device; throughput; response time

0 引 言

磁盘阵列存储系统控制软件的目的是把多个硬盘驱动器连接在一起协同工作,大大提高数据的存取速度,同时把硬盘存储系统的可靠性提高到接近无错的境界。这些"容错"冗余存储系统提供了高速的数据访问方式,同时保证极高的数据可靠性。其主要功能包括支持多种常用阵列级别:RAID0,RAID1,RAID10,RAID4,RAID5,特别支持双盘容错的RAID6[1]。高效的多级逻辑卷管理器提供更为灵活的数据存储方案。数据加密安全子系统保证数据的高安全性。支持2Gbps光纤通道主机接口和Ultra3 SCSI320设备通道接口。阵列状态监测及故障情况下自动在线重建数据。支持多种在线扩容方式。支持热插拔及热备份盘。同时可支持多条设备通道,容量高达TB级别。支持点对点、环、交换式光纤网络环境。使用磁盘阵列存储系统控制软件的优点包括提升系统整体性能、增加系统可靠性、增加储存容量、降低营运成本、增加系统扩展性及增加数据容错性。

RAID是一个复杂的系统,数据在从远程启动到实际的物理磁盘传输中,需要经过数据的分解、合并、校验等多种处理后才能最终完成[2]。尤其是数据需要在内核和用户以及块设备之间进行多次拷贝,这对RAID的性能影响也很大。目前在内核态实现数据的零拷贝的还很少。本设计实现的就是一种基于iSCSI的blockio的零拷贝的RAID控制器系统。经过试验表明,这种方法能够提高数据的吞吐量,减少系统数据的响应时间。

1 iSCSI概述

iSCSI,即Internet SCSI,是SCSI在IP网络上的传输协议,通过将SCSI协议在TCP/IP网络上进行传输实现了IP网络存储的功能。

收稿日期:2007-06-14
基金项目:国家973基础研究计划(2004CB318201)
作者简介:万亚平(1973-),男,湖南株洲人,博士研究生,研究方向为网络存储技术、海量存储技术;欧阳利军,副教授,研究方向为计算机网络技术。

1.1 iSCSI 的发展

iSCSI 是由 IBM 下属的两大研发机构——加利福尼亚 Almaden 和以色列 Haifa 研究中心共同开发的，是一个供硬件设备使用的可以在 IP 协议上层运行的 SCSI 指令集。简单地说，iSCSI 可以实现在 IP 网络上运行 SCSI 协议，利用已有的 IP 网络构成 SCSI 总线逻辑上的延伸。

从根本上说，iSCSI 协议是一种跨过 IP 网络来传输 SCSI 数据块的方法。它由 CISCO 和 IBM 等公司发起，并且得到了 IP 存储技术拥护者的大力支持。与光纤通道技术相比，iSCSI 比光纤通道有更好的可扩展发现和登录机制。iSCSI 可以利用 IP 网络中使用的身份验证和加密机制。iSCSI 的优势还在于可以利用现有的 IP 网络设施。

iSCSI 协议的第一个草案标准，由 Cisco 和 IBM 在 2000 年 1 月发布，目前已有一大批厂家生产了基于 iSCSI 的产品。iSCSI 允许 SCSI 命令通过 TCP/IP 协议传输，它整合了现有的存储协议 SCSI 和网络协议 TCP/IP 两种主流协议，实现了存储和网络的无缝融合。通过 iSCSI 协议，可以在标准以太网上访问存储设备和 SAN。TCP 会保证数据可靠性，管理网络拥塞，并采用中继重传策略来处理延迟。2003 年 2 月 11 日，IETF(Internet Engineering Task Force，国际互联网工程任务组)通过了 iSCSI 协议标准，并于 2004 年 4 月发布了 iSCSI RFC3720。

随着以 Gigabit Ethernet 为代表的高速局域网技术的迅猛发展，SCSI over IP，即实现在 IP 网络上的块级别存储，成为人们的研究兴趣。这方面早期的研究工作主要有：南加州大学 ISI(Information Science Institute)的 Netstation 项目在设备的接口方面选择了 SCSI，在网络传输上力挺 IP 网络；CMU 研究人员在 NASD(Network Attached Secure Disk)研究初期提出过另一个系统结构 Network SCSI Disk；IBM 的研究人员对 SCSI over IP 的各种接口选择作了研究评估，最终转到 SCSI over TCP 上。

2000 年，IBM，HP，Cisco 等公司的研究人员向 IETF 提交了一个标准提议，以图进行标准化。IETF 随后成立了名为 IPStorage(IPS)的工作组，以探求将现有协议如 SCSI、Fibre Channel 在 IP 网络上进行封装传输的切实实用的途径和方法。

该工作组的努力聚焦在传输以及相关的问题，具体有：

1)可靠的传输。要封装的目标协议如 SCSI 等需要可靠的传输，而 IP 网络本身是一个不可靠的网络，要缩小它们之间的差距，就必须在协议设计上下功夫。选择可靠的传输协议，考虑安全、错误的检测与恢复等重要机制；

2)资源的命名与发现机制。在 IP 存储的背景下，存储成为一种 Internet 广域范围内的资源，像之前 Internet 兴起的各种资源服务一样，存储资源需要一套方便灵活的命名与发现机制。

3)相关 MIB 的定义，以便通过 SNMP 进行监控管理。

IPS 工作组目前已经取得了一系列成果。在可靠传输方面，推出了三种封装协议：一种称为 Native IP-transport，即将 SCSI 直接通过 TCP/IP 进行封装，这个协议正是 iSCSI。另两种称为 Bridge IPtransport，它们的目的是将原来的 FC－SAN 通过 IP 网络连接起来，具体有采用隧道机制的 FCIP(Fibre Channel Internet Protocol)和采用网关到网关机制 iFCP(Internet Fibre Channel Protocol)两种解决方法。目前 iSCSI、FCIP 都已成为 RFC，iFCP 还处在草案阶段。在资源命名与发现方面，推出了 iSNS(Internet Storage Name Services)，以自动管理、配置、发现使用 iSCSI 和 iFCP 时的资源。IP 网络存储的安全机制、相关的 MIB 定义等方面也有积极的成果[3]。

1.2 iSCSI 的实现方式

iSCSI 使用 TCP/IP 协议在以太网上传输块存储的 SCSI 命令。

iSCSI 启动器通过以太网线连接发出 I/O 请求的主机。iSCSI 目标器直接连接存储设备。启动器封装 SCSI 命令和数据，使之能够被 TCP/IP 协议处理。网络远端的目标器解析收到的数据和命令。实现 iSCSI 的软件可以看作位于 TCP/IP 层上的协议层。这层协议既可以运行于主机上，也可以运行在主板的协处理器上。使用芯片实现也是一个途径。不同的实现方法决定 SAN 连接的性价比不同[4]。

目前实现 iSCSI 主要有以下三种方式：

1)纯软件方式：采用通用的以太网卡，iSCSI 和 TCP/IP 协议栈功能层都由主机 CPU 完成。由于采用的是标准的网卡，因此这种方式的硬件成本最低，但由于 iSCSI 和 TCP/IP 层功能都由主机 CPU 完成，通过该网卡的既有网络通信量又有存储通信量，随着这种通信量的增加，主机的运行开销就会大大增加，从而造成主机系统性能的下降，严重时还可能使主机成为系统的瓶颈。

2)智能 iSCSI 网卡实现方式：采用特定的智能网卡，iSCSI 层的功能由主机来完成，而 TCP/IP 协议栈功能由网卡来完成。和方式 1 相比，部分降低了主机的运行开销。

3)iSCSI HBA卡实现方式:采用主机总线适配器的方式,iSCSI层和TCP/IP协议栈功能均由该主机总线适配器来完成。对主机的CPU的需求最少,相对主机而言,就是一个标准的SCSI HBA,可以在各种操作系统平台上应用。

2 本设计中RAID系统的硬件架构配置

磁盘阵列系统控制软件的硬件架构使用MacroSource公司的HighPro1108系列的SATA RAID控制卡连接磁盘,每个SATA控制卡可以连接8个SATA磁盘,总共可以连接40个磁盘,理论上存储容量最大可以达到6TB;外部通道是主从通道,专门用来连接主机和阵列并进行命令和数据的传输,文中采用的是2Gbps数据传输率的光纤通道技术,理论上可以达到全双工400MB/s的持续数据传输率,可以满足底层通道的数据带宽的要求。具体使用Qlogic公司的QLA2310F光纤通道主机适配器,可以连接126个光纤设备。

其特点是:低CPU占用率;支持热插拔,在主机系统运行时就可安装或拆除光纤通道硬盘;可实现光纤和铜缆的连接;高带宽,在适宜的环境下,光纤通道是现有产品中速度最快的;通用性强;连接距离大,连接距离远远超出其它同类产品[5]。

磁盘阵列控制器CPU为Pentium D 2.80GHz,主要作用是实时读取前置处理机内存中的数据,分解为每个独立磁盘驱动器的子命令,并实时写入对应磁盘驱动器内。同时完成数据分块/重组、计算校验信息,以及Cache-Buffer管理等功能。在数据事后处理时,对于访盘命令,通过I/O调度模块完成对磁盘的并行访问。多条SATA总线各自通过HighPro1108适配器和PCI-E总线相连。每个HighPro1108在磁盘阵列中相当于一个串控制器,它集成了一个高性能SATA总线核心。充当串控制器的多个HighPro1108并行工作,执行调度程序交付的I/O命令。串内SATA总线上的多个磁盘驱动器通过多线程调度可同时响应各自盘上的I/O请求。4串SATA总线上的磁盘驱动器正交构成多个校验组。采用QLA2310F光纤通道适配卡作为阵列的外部设备接口,通过光纤通道总线与主机(或交换机)相连,从而可以充分利用光纤通道的高数据传输率和全双工的特点提高磁盘阵列的性能(如图1所示)。

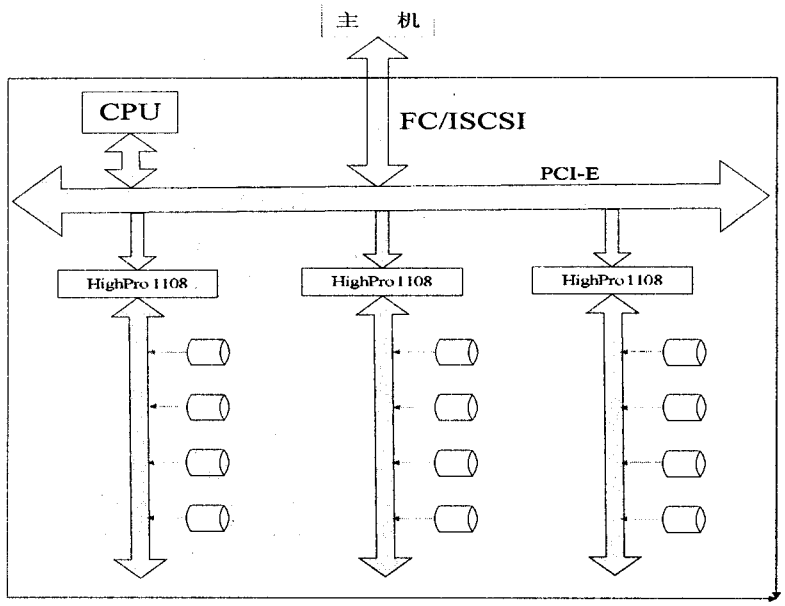

图1　磁盘阵列硬件组成结构图

3 磁盘阵列系统控制软件模块关系和数据流程

文中设计的RAID系统在ISCSI的Target端增加了解析功能模块,这有利于屏蔽掉通道的接口标准,可以无缝连接各种接口的底层设备。

图2即为本设计中各个主要的功能模块和数据流程关系。总控程序的功能是将各个磁盘上的存储空间按照一定的磁盘阵列级别进行组织,使得主机看来磁盘阵列相当于本地的一个物理驱动器,利用主机上的磁盘管理器来完成分区、格式化、安装文件系统等功能,给用户提供了方便熟悉的使用方式。总控程序整合了阵列配置和保存子模块、命令分解子模块、数据分块/重组子模块、校验计算子模块、数据重构/恢复子模块等模块于一体。

Target Driver与具体的硬件(FC接口卡)相关,因此,本部分的驱动代码一般由硬件厂商提供,正常情况下不需要修改。所以本设计不涉及这部分的具体内容。

Target Mid-level是连接Target驱动层与SCSI命令解析层的中间件。

SCSI CMD解析模块连接Target中间层与RAID总控模块。它也是本设计中所涉及的主要部分。尤其是其中的block_io部分,它应用了Linux2.6内核中最新定义的一系列数据结构,比如说bio结构体[6,7]。bio结构体代表的是I/O操作,它可以包括内存中的一个或多个页,相对于以前的buffer_head结构体,可以不进行不必要的数据块的分割。并且由于bio结构是轻

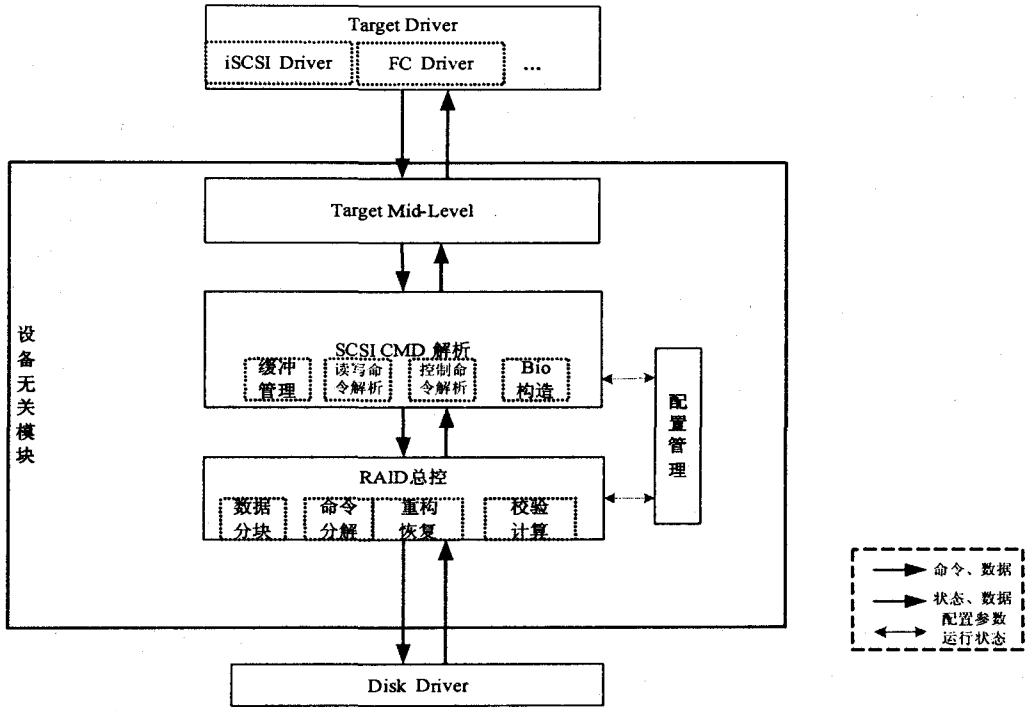

图 2 RAID 控制软件主要功能模块关系图

量级的[8],它描述的块可以不需要连续存储区。

利用 bio 结构体还有以下好处:

* bio 结构体很容易处理高端内存,因为它处理的是物理页而不是直接指针。

* bio 结构体既可以代表普通页 I/O,同时也可以代表直接 I/O(指那些不通过页高速缓存的 I/O 操作)。

* bio 结构体便于进行分散-集中(向量化的)块 I/O 操作,操作中的数据可取自多个物理页面。

* bio 结构体相比缓冲区头属于轻量级的结构体。因为它只需要包含块 I/O 操作所需要的信息就行了,不用包含与缓冲区本身相关的不必要信息。

4 性能测试与分析

本实验是在以上所述的硬件环境下,测试工具采用的 benchmark 是开源的 IOmeter。在测试时,主要测试了不同的块大小对本设计的性能影响以及和文件 I/O 进行比较的结果。

测试结果如图 3 所示。

从测试数据可以看出,采用 blockio 的设计改善了 fileio 下多次内存数据拷贝所带来的影响,因此数据吞吐量明显得到提高,在正常情况下,系统的峰值达到了 84MB/s。

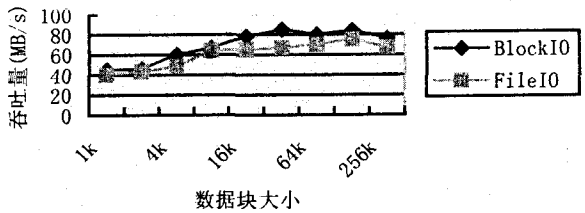

图 3 测试结果

参考文献:

[1] 谢长生,陆正武,谭志虎.一种提高 MD 读性能的方法[J].小型微型计算机系统,2004,25(7):1200-1203.

[2] 王 沛,韩耀伟.Linux 中 Software RAID 驱动程序的机制分析[J].小型微型计算机系统,2001,22(3):305-308.

[3] 冉春玉,陈才贤,胡恒莹,等.iscsi:网络存储的未来[J].微机发展,2004,14(8):17-20.

[4] 张南平,杨照芳,夏红霞.IPSAN 基于 IP 的存储区域网络[J].微机发展,2003,13(1):18-20.

[5] 张江陵,冯 丹.海量信息存储[M].北京:科学出版社,2003.

[6] Love R.Linux 内核设计与实现[M].北京:机械工业出版社,2006.

[7] Rubini A.Linux 设备驱动程序[M].中文版.北京:中国电力出版社,2000.

[8] Bovet D P,Cesati M.Understanding the Linux Kernel[M].[s.l.]:OReilly,2001.