

# 基于 FPGA 的可堆叠存储阵列设计与优化

王振升, 张 刚

(太原理工大学 信息工程学院, 山西 太原 030024)

**摘 要:** 提出一种新型的可堆叠磁盘存储阵列以及基于 FPGA 的直接控制存储阵列的存取技术, 在 FPGA 中实现磁盘控制器和阵列管理模块, 用于控制磁盘阵列读写; 为解决存储阵列的元数据和数据的读写访问问题, 设计了高速缓存和多通道读写控制机制, 处理模块将同一逻辑地址单元映射到  $q$  个通道中, SATA 控制模块将数据映射到通道内  $p$  个磁盘上, 实现了数据的并发读写访问. 实验结果表明, 可堆叠存储阵列不仅灵活高效, 而且可大幅提高整个存储系统的容量和带宽, 具有较大的应用价值.

**关键词:** 存储系统; FPGA 逻辑; 并行读写; 磁盘阵列; 可堆叠

**中图分类号:** TP334.5

**文献标识码:** A

**文章编号:** 1000-7180(2016)11-0001-04

## Design and Optimization of Stackable Storage Array Based on FPGA

WANG Zhen-sheng, ZHANG Gang

(College of Information Engineering, Taiyuan University of Technology, Taiyuan 030024, China)

**Abstract:** In order to improve the system bandwidth and storage capacity of the storage system, designing a stackable disk storage array model, and the controller and the disk array management module was implemented in the FPGA; To solve the problem of metadata and data, designing cache and multi-channel read and write control mechanism, processing module with a logic unit address mapping to the  $q$  channel, and SATA control module data mapping to channel  $p$  on the disk, realizing the parallel read and write access of the data. The experimental results show that the stackable storage array is efficient and flexible, and can greatly improve the storage capacity and bandwidth, and has great application value.

**Key words:** storage system; FPGA logic; parallelizing reads and writes; disk array; stackable

### 1 引言

在互联网高速发展的背景下,迫切需求高速、稳定且可动态配置的高速磁盘存储阵列. 目前存在的 7 种基本的 RAID 结构<sup>[1]</sup>, 可实现存取速度的提升和数据的冗余存储, 但基于软件的 RAID 在稳定性和吞吐能力上有待提高. SATA 协议是基于 ATA 协议硬盘厂商组成的委员会提出的用以解决高传输速率下的 ATA 信号电磁干扰问题的硬盘接口标准<sup>[2]</sup>, 目前专用的高速数据存储的 SATA 磁盘阵列控制器还没有实现, 虽有一些研究在 FPGA 上实现了 SATA 磁盘控制器, 但缺少基于 RAID 的 SATA

磁盘控制器高速并行存取的整体解决方案. 本设计提出了新型的高速并行可堆叠存储阵列的硬件结构, 并在 FPGA 内硬件实现基于 SATA 的磁盘阵列控制器和阵列控制管理模块, 着重研究了多通道模式下磁盘阵列的划分和管理及读写控制访问优化.

### 2 系统整体方案设计

传统系统通过一个运行在刀片服务器上的驱动程序连接物理层芯片<sup>[3]</sup>. 驱动程序的任务是将文件系统逻辑上的“块 I/O”映射为存储介质物理上的“扇区 I/O”. 可堆叠存储阵列连接模式如图 1 所示, Marvell 物理层芯片连接到 FPGA 的一个通道控制模块, 由 FP-

GA 的阵列控制管理模块通过消息交换获得访问的块 I/O,由文件系统映射模块并发地将每个“块 I/O”映射为多个“扇区 I/O”,由校验和并行处理模块生成校验数据,并行驱动通道控制逻辑访问物理层芯片。

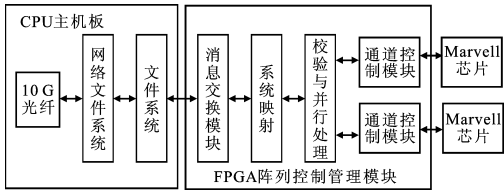


图 1 系统整体模块划分

2.1 系统架构

可堆叠存储阵列各模块设计如图 2 所示,主要包括万兆以太网接口、12.5 G 的 SATA 对外接口、PCIE×4 连接接口、SATA 磁盘控制器、Marvell 1 转 4 扩展接口和 40 个可热拔插磁盘。基于单个本地磁盘的访问是按磁头、柱面和扇区进行的,可堆叠磁盘阵列是由  $N$  个磁盘组成的磁盘阵列,为实现磁盘的快速定位和各磁盘功能划分,将多个磁盘按群、组、通道划分,不同功能的磁盘属于不同的群,实现同一功能的  $N$  个磁盘组沉的群可以划分为不同的组,同一组内串连在一起的磁盘称为一组,在系统上电时,系统读取磁盘分布的物理参数并进行初始化。系统初始化参数定义如表 1。

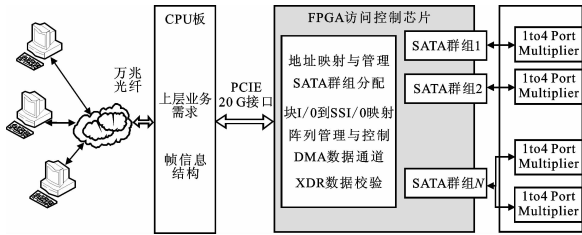


图 2 存储阵列模块设计

表 1 系统初始化参数

| 名称             | 长度 /Byte | 描述                                                                                |
|----------------|----------|-----------------------------------------------------------------------------------|
| 属性 Attr        | 1        | 类型标识(0X27)                                                                        |
| 群号 GRP         | 1        | 作用域 GRP                                                                           |
| 组号 SET         | 1        | SET 在阶段 1 为群内的组号,在阶段 2 为群内设备的名称(或目标地址);SET = 0Xff 表示该命令帧为群内广播方式                   |
| 命令 CMD         | 1        | 操作码                                                                               |
| 命令参数 Para      | 6        | 根据命令不同,参数 Para 格式需另外定义                                                            |
| 块 I/O 请求编号 BRQ | 1        | 所响应的 BI/O 请求编号,队列最多有 256 个 BI/O 请求(0~255),队列满时 CPU 不再处理新的块请求;1 个块请求是 1 个 SSI/O 序列 |
| 时间信息 Time      | 3        | CPU 端建立的时间戳                                                                       |
| 保留 RSV         | 2        | 保留                                                                                |

2.2 磁盘阵列划分

磁盘的读写访问依靠 SATA 通道控制模块实现,基本的读写访问是:对于确定的通道,按 DMA 方式读写同一群组内的  $n$  个磁盘的以某扇区地址作为起始地址的连续  $t$  个扇区的内容( $t > 0$ , 整数)。用  $q$  表示不同连接方式下 1 个逻辑单元包含的最小扇区数,用  $p$  表示每个通道容纳的硬盘个数;本系统在系统的逻辑单元“块”和存储介质的物理单元“扇区”之间引入超扇区 SS(Super Sector)的概念。1 个超扇区 SS 的尺寸  $n$  在系统初始化时确定, $n$  与通道数  $q$ ,每通道硬盘数  $p$ ,以及参数  $r$  有关,其关系由式(1) 决定。对于  $q$  个通道中位置相同的硬盘,其物理地址相同的  $q$  个扇区可以联合起来实施并行读写访问,这里将它们定义成 1 个扇区堆 SP(Sector Piles)。

初始化时需确定超扇区 SS 的参数  $r$ 。 $r = 0$  意味着 1 个超扇区 SS 的尺寸  $n$  为  $q$  个物理扇区(即 1 个扇区堆 SP); $r = 1$  表明 1 个 SS 的尺寸  $n$  为  $pq$  个物理扇区(由每个通道的  $p$  个硬盘中序号相同的  $p$  个扇区堆 SP 组成); $r > 1$  表明 1 个 SS 的尺寸  $n$  为  $rpq$  个扇区(由每个通道的  $p$  个硬盘中序号相同的连续  $r$  个扇区堆 SP 组成)。

若系统初始化时规定 1 个逻辑单元包含  $n$  个扇区,则  $n$  是乘积  $p!q$  的整数倍;逻辑单元的尺寸  $n$  与  $p \cdot q$  之间的关系由参数  $r$  确定:

$$\begin{cases} n = pr \cdot q & r = 0 \text{ 或 } 1 \\ n = r \cdot p \cdot q & r > 1 \end{cases} \quad (1)$$

参数  $q$  和  $p$  被连接方式确定,初始化时还需设置参数  $r$ ,从而确定 1 个逻辑单元包含的扇区个数  $n$ 。

按 DMA 方式将连续  $l(l > 0)$  个逻辑单元的内容,即  $l \times n$  个扇区的内容依次轮流读写到  $q$  个不同硬盘通道,为使负载平衡,具体读写方式如下:

在  $qk, qk + 1, qk + 2, \dots, qk + (q - 1)$  时( $k = 0, 1, \dots, l - 1$ ),读写访问的操作分别并行地针对第 0, 第 1, 第 2,  $\dots$ , 和第  $q - 1$  个通道执行;

具体到 1 个独立通道,对于  $r = 0$ :

每次读写个 1 硬盘的 1 个扇区,读写操作仅针对 1 个硬盘执行;

对于  $r \geq 1$ :

连续读写 1 个硬盘的  $r$  个扇区,读写 1 个硬盘连续  $r$  个扇区的访问操作分别依次针对该通道的第 0, 第 1,  $\dots$ , 和第  $p - 1$  个硬盘执行,每次总计读写  $rp$  个扇区。

2.3 读写访问控制

存储系统通过元数据来管理存储在磁盘上的数

据并完成高效访问<sup>[4]</sup>,主要有两类元数据:

(1)带有路径的文件名,用来建立和管理文件目录和文件名;

(2)逻辑存储单元的地址链表,用来实现块 I/O 的访问。

不同路径下的相同文件名可能对应着不同的文件,每个文件唯一对应一个逻辑存储单元的地址链表,文件系统通过文件名访问属于自己的地址链表。系统以块(Block)作为逻辑存储单元的地址索引,一个块由连续的  $m$  个超扇区(Super Sector)组成在初始化时规定一个逻辑存储单元“块”包含超扇区 SS 的数目  $m$ ,系统按照所定义的逻辑单元“块”编址,生成位图和逻辑“块”地址的链表。

### 3 模块原理与设计

单块硬盘的持续数据流读写速度最高 100 MB/s,一般能维持在 50~70 MB/s<sup>[5]</sup>,而随着数据量的不断提升和大数据的应用,对存取速度提出了更高要求,达到 400 MB/s、800 MB/s 甚至几 GB/s。采用可堆叠磁盘存储阵列是提升性能的一种关键技术,并且同时控制多路磁盘阵列需要磁盘阵列控制器,通过在 FPGA 上实现基于 SATA 硬盘控制器的磁盘阵列控制器和阵列管理模块是突破存取性能瓶颈的关键。

#### 3.1 SATA 磁盘控制器和阵列控制管理模块

本设计在 FPGA 内硬件实现 SATA 接口标准的磁盘阵列控制器,如图 3 所示,通过 Marvell 1 转 4 多路接口与磁盘相连,利用 FPGA 的逻辑资源,实现物理层、链路层、传输层和应用层<sup>[6]</sup>。链路层用于对物理层上的数据流进行数据解包并对传输层传送来的数据加控制字符并打包,对传输的数据进行 CRC 校验或计算,并完成加扰处理以避免电磁干扰,保证数据传输的正确性。

MFS(Message Frame Structure)信息帧结构是基于本设计提出的一种封装的接口协议,业务需求层将接收到的命令和数据在链路层封装成 MFS 结构,经过 PCIe 总线传输到 FPGA 逻辑层。MFS 分为命令帧、数据帧和状态帧,FPGA 内的磁盘控制器根据 MFS 帧头确定此数据的帧类型,元数据管理模块依据帧类型和帧内命令判定是否需要读取相关数据的元数据。业务需求层有新的业务需求是首先发送命令帧,元数据管理模块依据此信息读取存储在高速缓存或者磁盘阵列上的元数据以获取文件路径,通过读取文件路径找到文件并获得文件属性,

文件属性表示了文件所有者和相关的读、写、修改权限、文件的最后写改时间、文件大小、文件的存储位置等相关信息,依据文件属性判定相关操作是否能够执行,如果不能则返回失败标志;如果有相应的执行权限,则根据任务信息和任务执行情况执行相应的命令。

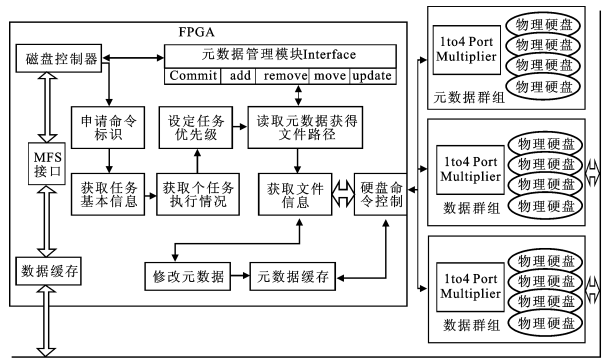


图3 存储系统硬件模块

#### 3.2 高速缓存

FPGA 接收到前端传输的数据需要先通过缓存机制解决存取的速度匹配问题,为提高系统存取效率,采用 FPGA 直接控制传输完成磁盘的连续 DMA 读写<sup>[7-8]</sup>,在 FPGA 内定义了两个 FIFO 分别用于 DMA 读和 DMA 写。2 个超扇区 I/O 队列互为副本,分别部署在 T2080 的软件和 FPGA 硬件中;T2080 软件负责超扇区 I/O 进入队列的操作,FP-GA 硬件负责队列的超扇区 I/O 响应以及响应结束后超扇区 I/O 移出队列的操作。超扇区 I/O 队列同步模块负责维护 2 个队列副本,确保数据一致性。超扇区 I/O 队列互斥模块负责超扇区 I/O 队列操作的独立性,即任一时刻只能发生一个针对队列的操作。数据缓冲管理模块包括超扇区 I/O 队列管理模块、写通道数据缓冲管理模块和读通道数据缓冲管理模块,读/写通道数据管理模块又分别由读/写通道数据缓冲区、环形队列控制逻辑和双缓存控制逻辑组成。数据缓冲管理模块受两个操作源的控制,一个只读操作源和一个只写操作源;写通道与读通道这两个数据缓冲管理模块完全相同,只是读写方向相反。

文件的访问基于元数据,为提高高速并行文件系统的并行性及存取速度,元数据存取单独占有一个磁盘群组,采用元数据与数据分离的管理结构,并在 FPGA 内的缓存区设立元数据缓存。如图 4 所示,元数据的操作主要包含代表具体的已安装文件的超级块对象、代表具体文件的索引节点对象、标识

路径的目录项对象和代表进程已打开文件的文件对象. 文件的关键信息以群组的归属权限包含在元数据中, 文件大小信息依据 FSD 固定分块信息封装成不同大小的块并重新组合. 缓存信息设定 AC(account time)值, 定时的同步更新元数据磁盘信息, 打开文件对象时查找元数据缓存信息, 基于局部性原理的高缓存命中率, 文件对元数据的访问实质上是访问高速缓存, 极大地提高了读取、写入与更新速度.

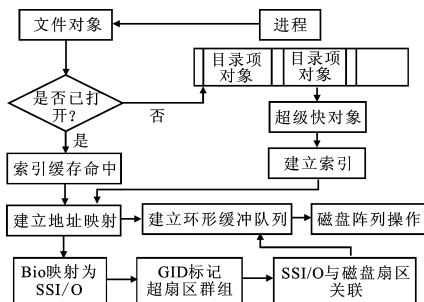


图4 元数据操作流程

### 3.3 多通道读写控制

图5中, 应用处理模块将1个逻辑地址单元(超扇区SS)映射到 $q$ 个通道(对连接方式0,  $q=4$ , 对连接方式1,  $q=8$ ); 而SATA通道控制模块将数据映射到通道内的 $p$ 个硬盘中(本系统 $1 \leq p \leq 4$ ). 对于硬盘节点 $p=4$ 的情况, 2位二进制数 $ADDR_{10}$ 用来表示通道内硬盘结点的编号. 将 $ADDR_{10}$ 部署到硬盘扇区地址的最低位, 实现通道内扇区统一编址. 通道内扇区地址的第0(b00 00)、4(b01 00)、8(b10 00)和0Xc(b11 00)号扇区分别对应于0号硬盘结点的0、1、2和3号扇区, 下划线00表示它们的通道内硬盘结点的编号(扇区统一地址最低2位 $ADDR_{10}$ )都是00; 将1(b00 01)、5(b01 01)、9(b10 01)和0Xd(b11 01)号扇区分别对应于1号硬盘结点的0、1、2和3号扇区, 依此类推, 下划线01表示它们的通道内硬盘结点的编号(扇区统一地址最低2位 $ADDR_{10}$ )都是01. 例如要将连续 $m=2^2=4$ 个扇区数据写入某通道, 起始扇区地址为0X23(b001000 11), 则第1个扇区的数据写入到3号硬盘结点( $ADDR_{10}=3$ )的0X08(b001000)号扇区, 第2个扇区的数据写入到0号硬盘结点( $ADDR_{10} \uparrow 1=00$ )的0X09(b001001)号扇区, 第3个扇区的数据写入到1号硬盘结点( $ADDR_{10} \uparrow =01$ )的0X09(b001001)号扇区, 第4个扇区的数据写入到2号硬盘结点( $ADDR_{10} \uparrow =$

10)的0X09(b001001)号扇区.

对于硬盘节点 $p=4$ 的情况, 如果要将连续 $m=2^{k+2}$ 个扇区数据写入某个通道, 只需在初始化时将扇区地址的第 $k+1$ 位和第 $k$ 位 $ADDR_{k+1,k}$ 定义成通道内硬盘结点的编号, 可将 $m=2^{k+2}$ 个连续扇区依次轮流写入相应的硬盘, 在增加带宽的同时实现了负载均衡.



图5 多通道控制管理模块

## 4 系统验证

本设计是用于集中式管理的高速网络存储的硬件组成部分, 采用 Freescale 的 T2080 作为 CPU 主机板, FPGA 板采用 Xilinx K7-325, 根据设计需求裁剪并配置 Linux 操作系统, 并将操作系统移植到 CPU 板, 在 Xilinx ISE 测试环境下搭建软硬件测试平台, 选用单盘容量为 80 G 的 SATA 硬盘组成单通道、4 通道和 8 通道的磁盘存储阵列. 在本实验室搭建局域网的测试环境, 上位机通过网络发送数据传输命令, 单次传输 10 GB 数据, 经测试单盘速度可达 156 MB/s, 4 通道的速度可达 431 MB/s, 8 通道可达 714 MB/s. 测试结果表明, 可堆叠存储阵列的设计可提高整个存取系统的带宽, 在提高存储系统容量的同时宽展了系统的吞吐能力, 为今后设计更大规模的存储阵列提供了设计范例.

## 5 结束语

本设计提出了新型的磁盘阵列架构, 在 FPGA 上基于 SATA 接口标准设计了磁盘存储控制器和阵列控制管理模块, 实现了多通道读写控制. 测试结果表明, 可堆叠存储阵列可大幅提高系统带宽和整个存储系统容量, 基于 FPGA 的可堆叠存储阵列最终可扩展为 40 路并行读写, 这为解决现有的存取速度受限、存储容量过小等问题提供了良好的解决方案, 具有较大的实用价值和广泛的应用前景.

## 参考文献:

[1] 张东. 大话存储—存储系统架构与底层原理极限剖析

(下转第9页)

了遗传加差分算法的云计算任务调度策略,改进的遗传算法对进化具有导向作用,明显减少搜索的盲目性。也可避免遗传算法的欺骗性问题,陷入局部最优解。通过实验仿真算法在成本,任务完成时间,负载均衡度上效果均好于遗传算法,Min-min,证明了本算法的有效性。实际中云计算的调度环境更为复杂,下一步会考虑任务间依赖关系,以及大任务量的调度,以及用户满意度等因素。

## 参考文献:

- [1] Mezma M, Melab N, Kessaci Y, et al. A parallel bi-objective hybrid metaheuristic for energy-aware scheduling for cloud computing systems[J]. Journal of Parallel & Distributed Computing, 2011, 71 (11): 1497-1508.
- [2] 刘鹏. 云计算[M]. 北京: 电子工业出版社, 2011.
- [3] 李建锋,彭舰. 云计算环境下基于改进遗传算法的任务调度算法[J]. 计算机应用, 2011, 31(1):184-186.
- [4] 熊聪聪,冯龙,陈丽仙,等. 云计算中基于遗传算法的任务调度算法研究[J]. 华中科技大学学报:自然科学版, 2012(Sup1):1-4.
- [5] 朱宇航,郑丽英,邬开俊. 基于改进 DE 的云计算任务调度算法[J]. 兰州交通大学学报, 2013, 54 (1): 101-106.

- [6] Chen S, Wu J, Lu Z. A cloud computing resource scheduling policy based on genetic algorithm with multiple fitness[C]// Computer and Information Technology, International Conference on. Chittagong, Bangladesh, IEEE, 2012:177-184.
- [7] Calheiros R N,Ranjan R. CloudSim:a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms[J]. Software-Practice and Experience,2011,41(1):23-50.
- [8] 李亚洲,郑晓军,张强等. 基于改进初始解的遗传算法的布局设计方法[J]. 计算机工程与应用,2013,49(8): 245-248.
- [9] Epitropakis M G,Tasoulis D K,Pavlidis N G,et al. Enhancing differential evolution utilizing proximity-based mutation operators[J]. IEEE Transactions on Evolutionary Computation,2011,15(1):67-98.

## 作者简介:

**葛君伟** 男,(1961-),博士,教授. 研究方向为云计算软件工程.

**孙方方(通讯作者)** 女,(1987-),硕士. 研究方向为云计算软件工程. E-mail:sunfangfang12@126.com.

**方义秋** 女,(1963-),博士,副教授. 研究方向为云计算数据库.

(上接第4页)

- [M]. 北京:清华大学出版社,2011:63-96.
- [2] Finisar. The path from 3Gb/s to SATA 6Gb/s;how to migrate current designs to the SATA revision 3.0 specification[EB/OL]. [2006-02-15]. <https://sata-io.org/sites/default/files/documents/SATA-6-Gbs-The-Path-from-3gbs-to-6gbs.pdf>.
- [3] 刘芬. 刀片服务器的发展[J]. 科技视界, 2014, 3 (24):97.
- [4] Daniel P Bovet, Marco Cesati. Understanding the Linux Kernel[M]. Cambridge:O'Reilly,2000.
- [5] Seagate Technology LLC. Barracuda 7200.11 serial ATA product manual revision B [M]. USA:Seagate,2008.
- [6] 欧阳科文. 基于FPGA的固态硬盘控制器设计[D]. 湖南:湖南大学,2009.
- [7] Wu wei, Su haibing. A high performance serial ATA

host controller[C]//Proc. of 2008 IEEE I on Computer Science and Software Engineering. [S. 1.]: IEEE Press, 2008: 18-23.

- [8] Louis Woods, Ken Oguro. Groundhog a serial ATA host bus adapter (HBA) for FPGAs [EB/OL]. [2016-02-19]. [http://people.inf.ethz.ch/lwoods/publications/Groundhog-A\\_Serial\\_ATA\\_Host\\_Bus\\_Adapter\\_\(HBA\)\\_for\\_FPGAs.pdf](http://people.inf.ethz.ch/lwoods/publications/Groundhog-A_Serial_ATA_Host_Bus_Adapter_(HBA)_for_FPGAs.pdf).

## 作者简介:

**王振升** 男,(1989-),硕士研究生. 研究方向为嵌入式系统及分布式存储. E-mail:tsuzs@163.com.

**张刚** 男,(1953-),教授,博士生导师. 研究方向为基于嵌入式系统的多媒体开发应用.