一种面向视频播放系统的 RAID 并行预取技术及实现

李 宇 张江陵 冯 丹

(华中科学技术大学计算机学院 武汉 430074)

(promisingmao@263.net)

摘 要 随着视频数字化技术的广泛应用,视频播放系统中的存储"瓶颈"也日益突出.在分析视频存储系统磁盘阵列的 I/O 调度算法和在实际应用中视频数据的特点的基础上,提出一种面向视频媒体服务的 RAID 并行预取实时调度算法,它利用未来数据的可预测性,预先将其从磁盘取入缓冲区,同时优化任务调度,使主机数据的回送和从磁盘预取数据同步执行,进一步提高了阵列的 I/O 吞吐率.用 I/O meter 测试结果证明,新算法具有很好的性能.

关键词 连续媒体流,磁盘阵列,VOD,并行预取,任务调度

中图法分类号 TP333.2

A PARALLEL PREFETCHING METHOD IN RAID ORIENTED TO MEDIA SERVER

LI Yu, ZHANG Jiang-Ling, and FENG Dan

(*Department of Computer Science, Huazhong University of Science and Technology, Wuhan* 430074)

Abstract With the popular application of the video digital technology, the storage bottleneck of the video service system is becoming increasingly serious. Based on the analysis of the I/O scheduling algorithm of RAID and the characteristics of the video data, a new parallel prefetching real-time dispatch method for video media service is designed. It forecasts the next batch of data requested by the host and fetches the data from the disks to the buffer in advance. At the same time, it optimizes the tasks dispatching, which enables the two tasks to be executed in parallel: the task of returning the requested data to the host and the task of prefetching data from disks. It is seen from the result by I/O meter that the new method improves the performance greatly.

Key words continuous media, RAID, VOD, parallel prefetching, task dispatch

1 引 言

磁盘阵列(RAID)是解决计算机系统中 I/O 瓶颈的一种有效方法,它利用并行技术将主机请求的 I/O 数据进行分块并交叉存放在不同的独立物理数据磁盘上,使多个磁盘能够并行读写,从而提高 I/O 处理速度.它的主要优点是:

(1) 增加了数据安全性;

(2) 增大了磁盘容量;

(3) 提供更高的执行效率.

这些优点使得它应用于广泛的领域,尤其是视

频播放系统. 视频播放系统中的数据流属于连续多媒体流, 并要求尽可能少的响应时间. 这种高质量的视频播放服务又对阵列性能提出了最大的挑战. 为了更好地满足这种需求, 必须不断地提高RAID的性能. 提高磁盘阵列的性能, 一方面要选用高效的软件系统支持环境, 如实时操作系统. 另一方面要改进磁盘阵列软件系统的实现方法. 当硬件系统确定时, 控制软件对系统的性能就起着至关重要的作用. 并行预取便是从软件系统角度来改善RAID的性能.

我们在常规阵列系统[1](未采用并行预取策略的阵列系统, 如图1所示)的基础上根据视频流的特点, 寻求合适的预取策略, 并加以改进. 在常规阵列系统中, 磁盘阵列将来自主机的访问请求分解成对应于多个磁盘的请求, 所有请求服务(以下称为任务G)完毕后, 向主机返回数据、状态、接收命令(以下称为任务R), 是完全串行的.

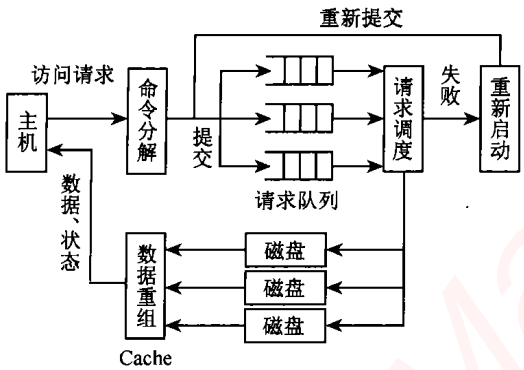

图1 常规陈列系统工作模型

2 并行预取技术

(1) 并行预取策略分析

假如当前主机要求返回的数据已经从磁盘中预先取出(放在缓冲区内), 那么任务R就可以不访问磁盘, 而直接将缓冲区的数据返回给主机, 同时返回读状态和接收下一条读命令. 而且当任务G从磁盘预取数据期间(由阵列串控制器控制执行), CPU处于空闲状态, 如果此时任务R要上传的数据在缓冲区"命中"的话, 便可和任务G并行执行, 这样势必更加减少响应时间. 这也就是本文所说的并行预取技术.

(2) 并行预取阵列系统的实现

下面分析视频播放系统中数据流的特点, 并根据它的特点分析面向此类数据流的预取策略.

视频文件的数据尺寸巨大, 如一部两小时左右的VCD质量的影片(码率为1.5Mbps), 容量达到1.5 GB左右. 因此视频文件一般存储在一个相对连续的空间, 而且数据访问一般是按顺序的.

根据这一特点, 预取数据时选择了与当前被请求数据相连续(地址连续)的下一批数据. 预取的时机一般有两种: 恒预取和不命中时预取. 恒预取是指当要访问某块时, 无论该块是否在Cache中, 都发预取命令. 不命中时预取则是只当访问到该块不命中时才发预取命令. 对主存Cache层次的模拟结果表明[2,3], 恒预取法使不命中率降低57%～58%, 而不命中预取法使不命中率降低30%～40%. 但前者所引起的Cache, 主存传输量的增加要比后者大得多. 因此本文所采用的预取时机在不考虑任务R和任务G的并行性时, 是不命中预取; 加上任务并行, 则在命中之后上传数据和状态时, 也同样进行预取. 这样将两种预取时机的优点融合起来了.

此外在具体实现中, 缓存有3个方面的问题需要考虑:

① 缓存结构, 为了使任务G和任务R达到并行, 同时也出于数据的一致性方面和减少缓存管理开销, 采用双缓存结构: 一个缓存用于存放调度程序从磁盘上回取的数据, 另一个缓存存放用于向主机输出流的数据;

② 缓存空间大小, 跟踪主机发至RAID的命令, 发现每条读命令读取数据的大小不会超过64KB, 因此缓存大小设为$64\times(n+1)$, n为当命令未命中时一次生成的预取命令数, 单位为KB, 这样能更有效地利用缓存空间;

③ 缓存替换算法, 由于视频播放这种应用具有很强的空间局部性, 适合选用FIFO算法, 而且这种算法实现起来也比较容易. 具体流程如图2所示.

3 性能分析

磁盘阵列系统的主要性能包括I/O响应时间和数传率. I/O响应时间是指主机发出I/O请求到I/O请求完成所需的时间. 为简化分析, 并行预取阵列系统(如图2所示)暂不考虑任务并行, 因此在常规阵列系统(如图1所示)和不考虑任务并行的预取阵列系统中, I/O响应时间都由以下几个部分组成: 排队等待时间、I/O服务时间以及磁盘适配器接口传输时间.

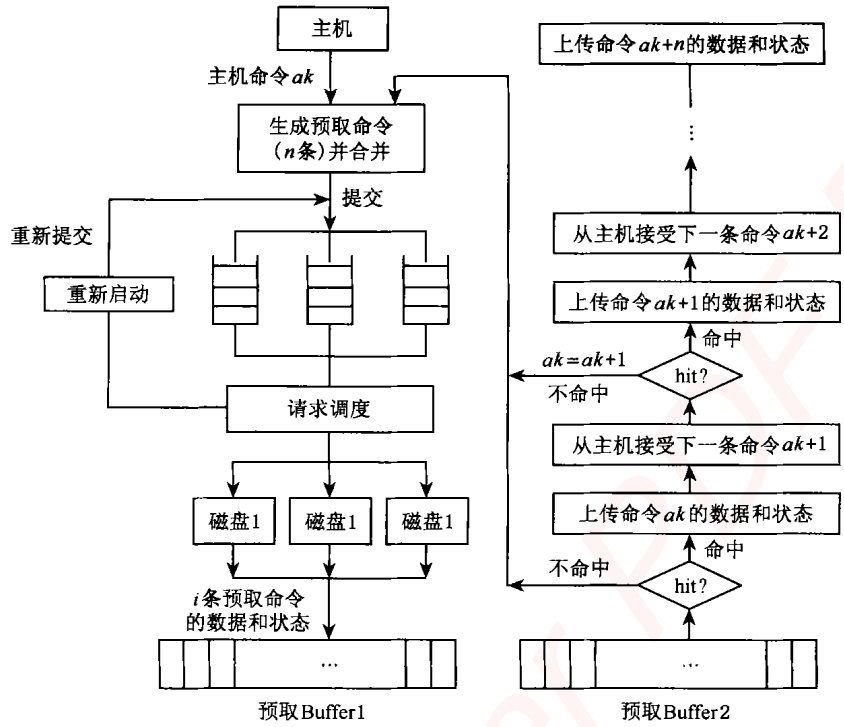

图 2　并行预取系统工作流程

常规阵列系统中, 我们用一组变量 $T_{响应}$, $T_{排队等待}$, T_{server}, $T_{数据传输}$ 分别表示 I/O 响应时间、排队等待时间、磁盘驱动器服务时间、接口传输时间 则有

$$T_{响应} = T_{排队等待} + T_{server} + T_{数据传输};$$

相应地, 预取系统(不考虑任务并行)中未命中命令的 I/O 响应时间表示为

$$T'_{响应} = T'_{排队等待} + T'_{server} + T'_{数据传输};$$

"命中"命令的 I/O 响应时间表示为

$$T''_{响应} = T''_{排队等待} + T''_{server} + T''_{数据传输}.$$

现在比较这两种系统中读取 $(n+1)\times D$ 字节大小的数据(为了简化分析, 设用户每次 I/O 请求大小为固定值 D 字节)的 I/O 响应时间

(1) 数据传输时间

设磁盘 Cache 到主存 Cache 的传输速率为 k 字节/秒; 从主存 Cache 到主机的传输速率为 m 字节/秒, 如图 3 所示 则 D 字节大小的数据传输时间(从磁盘 Cache 到主机)为 $D/m + D/k$ 秒(数据不命中)或 D/m 秒(数据命中).

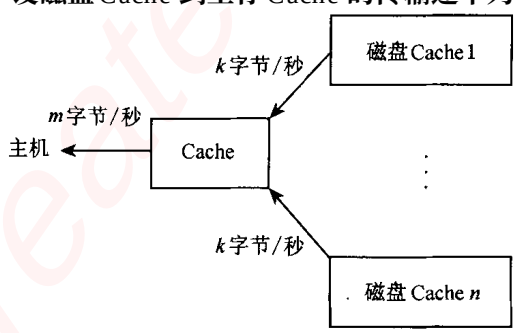

图 3　磁盘 Cache 与主存 Cache

(2) 驱动器服务时间[4]

磁盘驱动服务时间又由寻道时间(S_d)、旋转延迟(R_d)和数据传输时间(T_d)组成 T_{server} 表示磁盘驱动器服务时间 则有

$$T'_{server} = S_d + R_d + T_d.$$

为简单起见, 我们只讨论在同一磁道的扇区操作情况[3]. 从磁盘中读取连续的数据块 1 和数据块 2 所需时间的理论值 $T = t_1 + t_2 + t_3$; 其中 t_1 为读扇区 1 的旋转延迟; t_2 为读扇区 1 的数据传输时间; t_3 为读扇区 2 的数据传输时间 而实际上, 一块数据传输完毕, 启动第 2 块数据要做许多工作, 包括中断响应、磁头重新定位等 由于软件开销的存在, 使得一次将 $n+1$ 块地址连续的数据读出比分 $n+1$ 次去读节约时间 有

$$T'_{server} < (n+1) \times T_{server}.$$

若命令"命中", 这只要将 Cache 中的数据直接传送给主机, 有: $T''_{server} = 0$.

(3) 排队等待时间[5]

若主机 I/O 请求按泊松分布到达, 请求到达率为 λ, 假设每次主机 I/O 请求的平均磁盘驱动器服

务时间为 T (且 $T = T_{server} + T_{数据传输}$), T 的方差为 δ^2, 那么根据M/G/1 排队模型[3], 平均排队等待时间 $T_{排队} = (\lambda\sigma^2 + \lambda T^2)/2(1-\lambda T)$, $\lambda T \leq 1$; 排队等待时间 $T_{排队}$ 与 I/O 服务时间 T 成正比, 当 I/O 服务时间增大时排队等待时间也相应地增大.

(4) I/O 响应时间比较

要读取 $(n+1)D$ 大小的数据, 需要用户发出 $n+1$ 次的 I/O 请求. 常规阵列系统中, 设常规阵列系统中, 这 $n+1$ 次 I/O 请求响应时间之和 $T_{常规}$ 为:

$$T_{常规} = (n+1)T_{响应} = (n+1)(T_{排队} + T_{server}) + (m+k)(n+1)D,$$

在预取阵列系统中, 设这 $n+1$ 次 I/O 请求响应时间之和为 $T_{预取}$, 设这 $n+1$ 条命令中命中 I 条命令 $(0 \leq I \leq n)$, 则 $T_{预取}$ 为:

$$T_{预取} = (n+1-I)T'_{响应} + IT''_{响应} = (n+1-I)[T'_{排队} + T'_{server} + k(n+1)D] + m(n+1)D + IT''_{排队},$$

令 $\Delta t = T_{常规} - T_{预取}$, $\Delta t > 0$ 说明预取系统比常规系统好, 且 Δt 越大, 说明预取系统性能越好; $\Delta t < 0$ 说明常规系统比预取系统好.

显然, 当 $I = 0$, 即一条命令都没命中时, Δt 最小, 预取系统性能最差.

当 $I = n$ 时, 即除第 1 条命令外其余 n 条命令全部命中, Δt 最大, 预取系统性能最好.

当 $0 < I < n$ 时, 即有 $(n+1-I)$ 条命令要取出比本身所需数据多 n 倍的数据, 其中有 $L = (n+1-I) \times (n+1)D - I \times D$ 大小的数据为"废数据". L 越少, 也即在 n 固定的情况下, I 越大, 浪费时间越少. 另外 I 条命令(命中)的数据由于已经被预取到 Cache 中, 节约时间为 $I \times T_{server} + I \times D \times K$, 显然 I 越大, 节约的时间越多. 因此有结论: n 为定值的情况下, I 越大, 浪费的时间越少, 预取系统的性能越好. 与此同时 n 的取值也至关重要, n 越大, 有可能废数据越多, 但另外一方面 n 越大, T'_{server} 与 $(n+1) \times T_{sever}$ 的差值越大, 节约的时间越多.

并行预取系统(考虑任务 G 和任务 R 并行的预取系统)的性能显然还要优于上面所分析的预取系统(为简化分析而不考虑任务并行的预取系统).

基于上面的讨论, 我们采用 I/O meter 对并行预取系统和常规系统进行测试比较.

4 测试结果

在 pSOSystem 实时操作系统环境下, 用 I/O meter 来模拟视频服务环境(I/O meter 设置: Read 为 100%, Sequential 为 100%, 其余为缺省值) 对 RAID0(磁盘驱动器数: 3) 进行测试. 图 4 是 Trans Request Size 为 1 GB 时, 预取命令数不同, I/O 数传率的比较. 图 5 是不同的 Trans Request Size 设置, 这两种策略的性能比较.

由测试结果可知生成预取命令数越多, I/O 数传率越高. 且并行预取策略在读连续数据方面较常规策略有很大的优势.

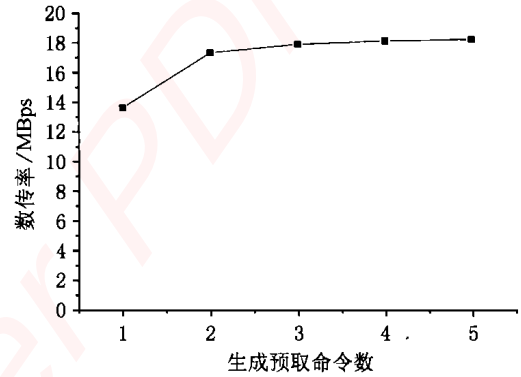

图 4 预取命令数不同, I/O 数传率的比较

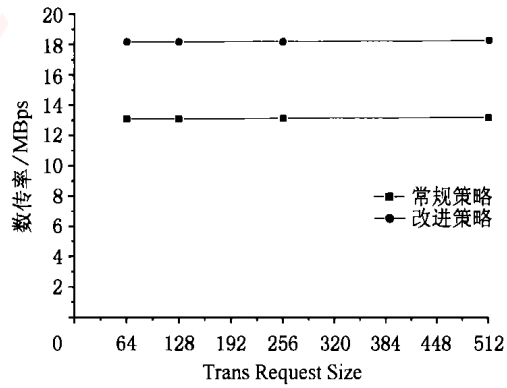

图 5 常规策略和预取策略数传率比较

5 总 结

预取就在于如何确定预取数据、预取策略和 Cache 策略如何配合. 生成的预取命令数不同(也就是预取数据量不同), 预取效果也不同. 理论上, 生成的预取指令数越多, 则可合并的命令数越多、预取效果越好, 可以获得更高的 I/O 数传率和更小的 I/O 响应时间, 但如果预取策略不适合或者"命中"率小于某个值, Cache 的缺失延迟又是造成有效速度远小于峰值的主要原因, 因为它会造成无用数据在磁盘和 Buffer 之间的频繁传送, 浪费系统时间, 反而带来更大的访问等待时间. 而且如果预取策略所增加的软件开销过大, 预取所带来的优势被其抵消, 就

有可能不仅不能给系统带来高性能,反而使系统性能下降 因此预取的数据和数据量要根据不同工作环境而调整,达到一个最佳值

参 考 文 献

1 陈鹏,张江陵,冯丹. 基于磁盘阵列的多队列实时I/O调度算法 华中理工大学学报, 1999, 27(4): 1～2
(Chen Peng, Zhang Jiangling, Feng Dan. Multi-queue real-time I/O scheduling algorithm based on disk array. Journal of Huazhong University of Science & Technology (in Chinese), 1999, 27(4): 1～2)

2 陈琼 并行存储系统的集成技术研究[博士论文]. 华中理工大学,武汉, 2000
(Chen Qiong. Research on integration technology of parallel storage system [Ph D dissertation] (in Chinese). Huazhong University of Science & Technology, Wuhan, 2000)

3 E B James, J F Michael. Reducing cache miss rates using prediction caches. Computer System Laboratory, Stanford University, Tech Rep: CSL-TR-96-707, 1996

4 周欣荣 面向盘阵列底层操作的优化控制研究[博士论文]. 华中理工大学,武汉, 1997
(Zhou Xinrong. Low level parallel operating in disk array [Ph D dissertation] (in Chinese). Huazhong University of Science & Technology, Wuhan, 1997)

5 罗荣桂 排队模型及其应用. 武汉: 华中理工大学出版社, 1990
(Luo Guirong. Queue pattern and its application (in Chinese). Wuhan: Huazhong University of Science & Technology Press, 1990)

李 宇 女, 1977年生, 硕士, 目前主要从事外存储并行性研究

张江陵 男, 1930年生, 教授, 目前主要从事信息存储技术研究

冯 丹 女, 1970年生, 教授, 目前主要从事海量存储系统、网络存储系统技术研究

嵌入式资源免费下载

总线协议：

1. 基于 PCIe 驱动程序的数据传输卡 DMA 传输
2. 基于 PCIe 总线协议的设备驱动开发
3. CANopen 协议介绍
4. 基于 PXI 总线 RS422 数据通信卡 WDM 驱动程序设计
5. FPGA 实现 PCIe 总线 DMA 设计
6. PCI Express 协议实现与验证
7. VPX 总线技术及其实现
8. 基于 Xilinx FPGA 的 PCIE 接口实现
9. 基于 PCI 总线的 GPS 授时卡设计
10. 基于 CPCI 标准的 6U 信号处理平台的设计
11. USB30 电路保护
12. USB30 协议分析与框架设计
13. USB 30 中的 CRC 校验原理及实现
14. 基于 CPLD 的 UART 设计
15. IPMI 在 VPX 系统中的应用与设计
16. 基于 CPCI 总线的 PMC 载板设计
17. 基于 VPX 总线的工件台运动控制系统研究与开发
18. PCI Express 流控机制的研究与实现
19. UART16C554 的设计
20. 基于 VPX 的高性能计算机设计
21. 基于 CAN 总线技术的嵌入式网关设计
22. Visual C 串行通讯控件使用方法与技巧的研究
23. IEEE1588 精密时钟同步关键技术研究
24. GPS 信号发生器射频模块的一种实现方案
25. 基于 CPCI 接口的视频采集卡的设计
26. 基于 VPX 的 3U 信号处理平台的设计
27. 基于 PCI Express 总线 1394b 网络传输系统 WDM 驱动设计
28. AT89C52 单片机与 ARINC429 航空总线接口设计
29. 基于 CPCI 总线多 DSP 系统的高速主机接口设计
30. 总线协议中的 CRC 及其在 SATA 通信技术中的应用
31. 基于 FPGA 的 SATA 硬盘加解密控制器设计
32. Modbus 协议在串口通讯中的研究及应用
33. 高可用的磁盘阵列 Cache 的设计和实现
34. RAID 阵列中高速 Cache 管理的优化

35. 一种新的基于 RAID 的 CACHE 技术研究与实现
36. 基于 PCIE-104 总线的高速数据接口设计
37. 基于 VPX 标准的 RapidIO 交换和 Flash 存储模块设计
38. 北斗卫星系统在海洋工程中的应用
39. 北斗卫星系统在远洋船舶上应用的研究
40. 基于 CPCI 总线的红外实时信号处理系统
41. 硬件实现 RAID 与软件实现 RAID 的比较
42. 基于 PCI Express 总线系统的热插拔设计
43. 基于 RAID5 的磁盘阵列 Cache 的研究与实现
44. 基于 PCI 总线的 MPEG2 码流播放卡驱动程序开发
45. 基于磁盘异或引擎的 RAID5 小写性能优化
46. 基于 IEEE1588 的时钟同步技术研究
47. 基于 Davinci 平台的 SD 卡读写优化
48. 基于 PCI 总线的图像处理及传输系统的设计
49. 串口和以太网通信技术在油液在线监测系统中的应用
50. USB30 数据传输协议分析及实现
51. IEEE 1588 协议在工业以太网中的实现
52. 基于 USB30 的设备自定义请求实现方法
53. IEEE1588 协议在网络测控系统中的应用
54. USB30 物理层中弹性缓冲的设计与实现
55. USB30 的高速信息传输瓶颈研究
56. 基于 IPv6 的 UDP 通信的实现
57. 一种基于 IPv6 的流媒体传送方案研究与实现
58. 基于 IPv4-IPv6 双栈的 MODBUS-TCP 协议实现
59. RS485CAN 网关设计与实现
60. MVB 周期信息的实时调度
61. RS485 和 PROFINET 网关设计
62. 基于 IPv6 的 Socket 通信的实现
63. MVB 网络重复器的设计
64. 一种新型 MVB 通信板的探究
65. 具有 MVB 接口的输入输出设备的分析
66. 基于 STM32 的 GSM 模块综合应用
67. 基于 ARM7 的 MVB CAN 网关设计
68. 机车车辆的 MVB CAN 总线网关设计
69. 智能变电站冗余网络中 IEEE1588 协议的应用
70. CAN 总线的浅析 CANopen 协议
71. 基于 CANopen 协议实现多电机系统实时控制
72. 以太网时钟同步协议的研究
73. 基于 CANopen 的列车通信网络实现研究
74. 基于 SJA1000 的 CAN 总线智能控制系统设计
75. 基于 CANopen 的运动控制单元的设计
76. 基于 STM32F107VC 的 IEEE 1588 精密时钟同步分析与实现

77. 分布式控制系统精确时钟同步技术
78. 基于 IEEE 1588 的时钟同步技术在分布式系统中应用
79. 基于 SJA1000 的 CAN 总线通讯模块的实现
80. 嵌入式设备的精确时钟同步技术的研究与实现
81. 基于 SJA1000 的 CAN 网桥设计
82. 基于 CAN 总线分布式温室监控系统的设计与实现
83. 基于 DSP 的 CANopen 通讯协议的实现
84. 基于 PCI9656 控制芯片的高速网卡 DMA 设计
85. 基于以太网及串口的数据采集模块设计
86. MVB1 类设备控制器的 FPGA 设计
87. MVB 接口彩色液晶显示诊断单元的显示应用软件设计
88. IPv6 新型套接字的网络编程剖析
89. 基于规则的 IPv4 源程序到 IPv6 源程序的移植方法
90. MVB 网络接口单元的 SOC 解决方案
91. 基于 IPSec 协议的 IPv6 安全研究
92. 具有 VME 总线的车载安全计算机 MVB 通信板卡
93. SD 卡的传输协议和读写程序
94. 基于 SCTP 的 TLS 应用
95. 基于 IPv6 的静态路由实验设计
96. 基于 MVB 的地铁列车司机显示系统研究
97. 基于参数优化批处理的 TLS 协议
98. SSD 数据结构与算法综述
99. 大容量 NAND Flash 文件系统中的地址映射算法研究
100. 基于 MVB 总线的动车组门控系统的设计与仿真研究
101. 多功能车辆总线 MVB 控制
102. 基于 LabVIEW 的 MVB 和 WTB 帧解码方法
103. 基于双 FPGA 的 MVB 通用接口研制
104. RAID 中 Cache 的设计与实现
105. RAID 及并行预取技术分析
106. RAID 系统中 RAID 级别的具体实现算法
107. 一种基于超大容量 Cache 的 VOD 系统

VxWorks:

1. 基于 VxWorks 的多任务程序设计
2. 基于 VxWorks 的数据采集存储装置设计
3. Flash 文件系统分析及其在 VxWorks 中的实现
4. VxWorks 多任务编程中的异常研究

5. VxWorks 应用技巧两例
6. 一种基于 VxWorks 的飞行仿真实时管理系统
7. 在 VxWorks 系统中使用 TrueType 字库
8. 基于 FreeType 的 VxWorks 中文显示方案
9. 基于 Tilcon 的 VxWorks 简单动画开发
10. 基于 Tilcon 的某武器显控系统界面设计
11. 基于 Tilcon 的综合导航信息处理装置界面设计
12. VxWorks 的内存配置和管理
13. 基于 VxWorks 系统的 PCI 配置与应用
14. 基于 MPC8270 的 VxWorks BSP 的移植
15. Bootrom 功能改进经验谈
16. 基于 VxWorks 嵌入式系统的中文平台研究与实现
17. VxBus 的 A429 接口驱动
18. 基于 VxBus 和 MPC8569E 千兆网驱动开发和实现
19. 一种基于 vxBus 的 PPC 与 FPGA 高速互联的驱动设计方法
20. 基于 VxBus 的设备驱动开发
21. 基于 VxBus 的驱动程序架构分析
22. 基于 VxBus 的高速数据采集卡驱动程序开发
23. Vxworks 下的冗余 CAN 通讯模块设计
24. WindML 工业平台下开发 S1d13506 驱动及显示功能的实现
25. WindML 中 Mesa 的应用
26. VxWorks 下图形用户界面开发中双缓冲技术应用
27. VxWorks 上的一种 GUI 系统的设计与实现
28. VxWorks 环境下 socket 的实现
29. VxWorks 的 WindML 图形界面程序的框架分析
30. VxWorks 实时操作系统及其在 PC104 下以太网编程的应用
31. 实时操作系统任务调度策略的研究与设计
32. 军事指挥系统中 VxWorks 下汉字显示技术
33. 基于 VxWorks 实时控制系统中文交互界面开发平台
34. 基于 VxWorks 操作系统的 WindML 图形操控界面实现方法
35. 基于 GPU FPGA 芯片原型的 VxWorks 下驱动软件开发
36. VxWorks 下的多串口卡设计
37. VxWorks 内存管理机制的研究
38. T9 输入法在 Tilcon 下的实现
39. 基于 VxWorks 的 WindML 图形界面开发方法
40. 基于 Tilcon 的 IO 控制板可视化测试软件的设计和实现
41. 基于 VxWorks 的通信服务器实时多任务软件设计
42. 基于 VXWORKS 的 RS485MVB 网关的设计与实现
43. 实时操作系统 VxWorks 在微机保护中的应用
44. 基于 VxWorks 的多任务程序设计及通信管理
45. 基于 Tilcon 的 VxWorks 图形界面开发技术
46. 嵌入式图形系统 Tilcon 及应用研究

47. 基于 VxWorks 的数据采集与重演软件的图形界面的设计与实现
48. 基于嵌入式的 Tilcon 用户图形界面设计与开发
49. 基于 Tilcon 的交互式多页面的设计
50. 基于 Tilcon 的嵌入式系统人机界面开发技术
51. 基于 Tilcon 的指控系统多任务人机交互软件设计
52. 基于 Tilcon 航海标绘台界面设计
53. 基于 Tornado 和 Tilcon 的嵌入式 GIS 图形编辑软件的开发
54. VxWorks 环境下内存文件系统的应用
55. VxWorks 下的多重定时器设计
56. Freescale 的 MPC8641D 的 VxWorks BSP
57. VxWorks 实验五[时间片轮转调度]
58. 解决 VmWare 下下载大型工程.out 出现 WTX Error 0x100de 的问题
59. 基于 VxWorks 系统的 MiniGUI 图形界面开发
60. VxWorks BSP 开发中的 PCI 配置方法
61. VxWorks 在 S3C2410 上的 BSP 设计
62. VxWorks 操作系统中 PCI 总线驱动程序的设计与实现
63. VxWorks 概述
64. 基于 AT91RM9200 的 VxWorks END 网络驱动开发
65. 基于 EBD9200 的 VxWorks BSP 设计和实现
66. 基于 VxWorks 的 BSP 技术分析
67. ARM LPC2210 的 VxWorks BSP 源码
68. 基于 LPC2210 的 VxWorks BSP 移植
69. 基于 VxWorks 平台的 SCTP 协议软件设计实现
70. VxWorks 快速启动的实现方法[上电到应用程序 1 秒]
71. VxWorks 下 SL811HS 的 Host 驱动源码

Linux：

1. Linux 程序设计第三版及源代码
2. NAND FLASH 文件系统的设计与实现
3. 多通道串行通信设备的 Linux 驱动程序实现
4. Zsh 开发指南-数组
5. 常用 GDB 命令中文速览
6. 嵌入式 C 进阶之道
7. Linux 串口编程实例
8. 基于 Yocto Project 的嵌入式应用设计
9. Android 应用的反编译
10. 基于 Android 行为的加密应用系统研究
11. 嵌入式 Linux 系统移植步步通

12. 嵌入式 CC++语言精华文章集锦
13. 基于 Linux 的高性能服务器端的设计与研究
14. S3C6410 移植 Android 内核
15. Android 开发指南中文版
16. 图解 Linux 操作系统架构设计与实现原理（第二版）
17. 如何在 Ubuntu 和 Linux Mint 下轻松升级 Linux 内核
18. Android 简单 mp3 播放器源码
19. 嵌入式 Linux 系统实时性的研究
20. Android 嵌入式系统架构及内核浅析
21. 基于嵌入式 Linux 操作系统内核实时性的改进方法研究
22. Linux TCP IP 协议详解
23. Linux 桌面环境下内存去重技术的研究与实现
24. 掌握 Android 7.0 新增特性 Quick Settings
25. Android 应用逆向分析方法研究
26. Android 操作系统的课程教学
27. Android 智能手机操作系统的研究
28. Android 英文朗读功能的实现
29. 基于 Yocto 订制嵌入式 Linux 发行版
30. 基于嵌入式 Linux 的网络设备驱动设计与实现
31. 如何高效学习嵌入式
32. 基于 Android 平台的 GPS 定位系统的设计与实现
33. LINUX ARM 下的 USB 驱动开发
34. Linux 下基于 I2C 协议的 RTC 驱动开发
35. 嵌入式下 Linux 系统设备驱动程序的开发
36. 基于嵌入式 Linux 的 SD 卡驱动程序的设计与实现
37. Linux 系统中进程调度策略
38. 嵌入式 Linux 实时性方法
39. 基于实时 Linux 计算机联锁系统实时性分析与改进
40. 基于嵌入式 Linux 下的 USB30 驱动程序开发方法研究
41. Android 手机应用开发之音乐资源播放器
42. Linux 下以太网的 IPv6 隧道技术的实现
43. Research and design of mobile learning platform based on Android
44. 基于 linux 和 Qt 的串口通信调试器调的设计及应用
45. 在 Linux 平台上基于 QT 的动态图像采集系统的设计
46. 基于 Android 平台的医护查房系统的研究与设计
47. 基于 Android 平台的软件自动化监控工具的设计开发
48. 基于 Android 的视频软硬解码及渲染的对比研究与实现
49. 基于 Android 移动设备的加速度传感器技术研究
50. 基于 Android 系统振动测试仪研究
51. 基于缓存竞争优化的 Linux 进程调度策略
52. Linux 基于 W83697 和 W83977 的 UART 串口驱动开发文档
53. 基于 AT91RM9200 的嵌入式 Linux 系统的移植与实现

54. 路由信息协议在 Linux 平台上的实现
55. Linux 下 IPv6 高级路由器的实现
56. 基于 Android 平台的嵌入式视频监控系统设计
57. Redhat linux 教程
58. 一种 Linux 平台下校园网服务器集群实现方案

Windows CE:

1. Windows CE.NET 下 YAFFS 文件系统 NAND Flash 驱动程序设计
2. Windows CE 的 CAN 总线驱动程序设计
3. 基于 Windows CE.NET 的 ADC 驱动程序实现与应用的研究
4. 基于 Windows CE.NET 平台的串行通信实现
5. 基于 Windows CE.NET 下的 GPRS 模块的研究与开发
6. win2k 下 NTFS 分区用 ntldr 加载进 dos 源代码
7. Windows 下的 USB 设备驱动程序开发
8. WinCE 的大容量程控数据传输解决方案设计
9. WinCE6.0 安装开发详解
10. DOS 下仿 Windows 的自带计算器程序 C 源码
11. G726 局域网语音通话程序和源代码
12. WinCE 主板加载第三方驱动程序的方法
13. WinCE 下的注册表编辑程序和源代码
14. WinCE 串口通信源代码
15. WINCE 的 SD 卡程序[可实现读写的源码]
16. 基于 WinCE 的 BootLoader 研究
17. Windows CE 环境下无线网卡的自动安装
18. 基于 Windows CE 的可视电话的研究与实现
19. 基于 WinCE 的嵌入式图像采集系统设计
20. 基于 ARM 与 WinCE 的掌纹鉴别系统
21. DCOM 协议在网络冗余环境下的应用
22. Windows XP Embedded 在变电站通信管理机中的应用
23. XPE 在多功能显控台上的开发与应用
24. 基于 Windows XP Embedded 的 LKJ2000 仿真系统设计与实现
25. 虚拟仪器的 Windows XP Embedded 操作系统开发
26. 基于 EVC 的嵌入式导航电子地图设计
27. 基于 XPEmbedded 的警务区 SMS 指挥平台的设计与实现
28. 基于 XPE 的数字残币兑换工具开发
29. Windows CENET 下 ADC 驱动开发设计
30. Windows CE 下 USB 设备流驱动开发与设计
31. Windows 驱动程序设计

32. 基于 Windows CE 的 GPS 应用
33. 基于 Windows CE 下大像素图像分块显示算法的研究
34. 基于 Windows CE 的数控软件开发与实现
35. NAND FLASH 在 WINCENET 系统中的应用设计

PowerPC：

1. Freescale MPC8536 开发板原理图
2. 基于 MPC8548E 的固件设计
3. 基于 MPC8548E 的嵌入式数据处理系统设计
4. 基于 PowerPC 嵌入式网络通信平台的实现
5. PowerPC 在车辆显控系统中的应用
6. 基于 PowerPC 的单板计算机的设计
7. 用 PowerPC860 实现 FPGA 配置
8. 基于 MPC8247 嵌入式电力交换系统的设计与实现
9. 基于设备树的 MPC8247 嵌入式 Linux 系统开发
10. 基于 MPC8313E 嵌入式系统 UBoot 的移植
11. 基于 PowerPC 处理器 SMP 系统的 UBoot 移植
12. 基于 PowerPC 双核处理器嵌入式系统 UBoot 移植
13. 基于 PowerPC 的雷达通用处理机设计
14. PowerPC 平台引导加载程序的移植
15. 基于 PowerPC 嵌入式内核的多串口通信扩展设计
16. 基于 PowerPC 的多网口系统抗干扰设计
17. 基于 MPC860T 与 VxWorks 的图形界面设计
18. 基于 MPC8260 处理器的 PPMC 系统
19. 基于 PowerPC 的控制器研究与设计
20. 基于 PowerPC 的模拟量输入接口扩展
21. 基于 PowerPC 的车载通信系统设计
22. 基于 PowerPC 的嵌入式系统中通用 IO 口的扩展方法
23. 基于 PowerPC440GP 型微控制器的嵌入式系统设计与研究
24. 基于双 PowerPC 7447A 处理器的嵌入式系统硬件设计
25. 基于 PowerPC603e 通用处理模块的设计与实现
26. 嵌入式微机 MPC555 驻留片内监控器的开发与实现
27. 基于 PowerPC 和 DSP 的电能质量在线监测装置的研制
28. 基于 PowerPC 架构多核处理器嵌入式系统硬件设计
29. 基于 PowerPC 的多屏系统设计
30. 基于 PowerPC 的嵌入式 SMP 系统设计
31. 基于 MPC850 的多功能通信管理器

32. 基于 MPC8640D 处理系统的技术研究
33. 基于双核 MPC8641D 处理器的计算机模块设计
34. 基于 MPC8641D 处理器的对称多处理技术研究
35. PowerPC 处理器原理
36. Freescale 的 P1020 参考设计原理图
37. Freescale T1040 参考设计板原理图

ARM：

1. 基于 DiskOnChip 2000 的驱动程序设计及应用
2. 基于 ARM 体系的 PC-104 总线设计
3. 基于 ARM 的嵌入式系统中断处理机制研究
4. 设计 ARM 的中断处理
5. 基于 ARM 的数据采集系统并行总线的驱动设计
6. S3C2410 下的 TFT LCD 驱动源码
7. STM32 SD 卡移植 FATFS 文件系统源码
8. STM32 ADC 多通道源码
9. ARM Linux 在 EP7312 上的移植
10. ARM 经典 300 问
11. 基于 S5PV210 的频谱监测设备嵌入式系统设计与实现
12. Uboot 中 start.S 源码的指令级的详尽解析
13. 基于 ARM9 的嵌入式 Zigbee 网关设计与实现
14. 基于 S3C6410 处理器的嵌入式 Linux 系统移植
15. CortexA8 平台的μC-OSⅡ及 LwIP 协议栈的移植与实现
16. 基于 ARM 的嵌入式 Linux 无线网卡设备驱动设计
17. ARM S3C2440 Linux ADC 驱动
18. ARM S3C2440 Linux 触摸屏驱动
19. Linux 和 Cortex-A8 的视频处理及数字微波传输系统设计
20. Nand Flash 启动模式下的 Uboot 移植
21. 基于 ARM 处理器的 UART 设计
22. ARM CortexM3 处理器故障的分析与处理
23. ARM 微处理器启动和调试浅析
24. 基于 ARM 系统下映像文件的执行与中断运行机制的实现
25. 中断调用方式的 ARM 二次开发接口设计
26. ARM11 嵌入式系统 Linux 下 LCD 的驱动设计
27. Uboot 在 S3C2440 上的移植
28. 基于 ARM11 的嵌入式无线视频终端的设计
29. 基于 S3C6410 的 Uboot 分析与移植

30. 基于 ARM 嵌入式系统的高保真无损音乐播放器设计
31. UBoot 在 Mini6410 上的移植
32. 基于 ARM11 的嵌入式 Linux NAND FLASH 模拟 U 盘挂载分析与实现
33. 基于 ARM11 的电源完整性分析
34. 基于 ARM S3C6410 的 uboot 分析与移植
35. 基于 S5PC100 移动视频监控终端的设计与实现
36. UBoot 在 AT91RM9200 上的移植简析
37. 基于工控级 AT91RM9200 开发板的 UBoot 移植分析
38. 基于 ARM11 和 Zigbee 的人员定位防丢器
39. 基于 NAND FLASH 的嵌入式系统启动速度的研究
40. μCOSⅡ在 ARM7 上的移植
41. 基于 ARM11 的嵌入式视频采集系统设计
42. 基于 ARM11 的视频监控系统设计
43. μC-OSII 在 LPC2210 上的移植研究

Hardware：

1. DSP 电源的典型设计
2. 高频脉冲电源设计
3. 电源的综合保护设计
4. 任意波形电源的设计
5. 高速 PCB 信号完整性分析及应用
6. DM642 高速图像采集系统的电磁干扰设计
7. 使用 COMExpress Nano 工控板实现 IP 调度设备
8. 基于 COM Express 架构的数据记录仪的设计与实现
9. 基于 COM Express 的信号系统逻辑运算单元设计
10. 基于 COM Express 的回波预处理模块设计
11. 基于 X86 平台的简单多任务内核的分析与实现
12. 基于 UEFI Shell 的 PreOS Application 的开发与研究
13. 基于 UEFI 固件的恶意代码防范技术研究
14. MIPS 架构计算机平台的支持固件研究
15. 基于 UEFI 固件的攻击验证技术研究
16. 基于 UEFI 的 Application 和 Driver 的分析与开发
17. 基于 UEFI 的可信 BIOS 研究与实现
18. 基于 UEFI 的国产计算机平台 BIOS 研究
19. 基于 UEFI 的安全模块设计分析
20. 基于 FPGA NiosⅡ的等精度频率计设计
21. 基于 FPGA 的 SOPC 设计
22. 基于 SOPC 基本信号产生器的设计与实现

23. 基于龙芯平台的 PMON 研究与开发
24. 基于 X86 平台的嵌入式 BIOS 可配置设计
25. 基于龙芯 2F 架构的 PMON 分析与优化
26. CPU 与 GPU 之间接口电路的设计与实现
27. 基于龙芯 1A 平台的 PMON 源码编译和启动分析
28. 基于 PC104 工控机的嵌入式直流监控装置的设计
29. GPGPU 技术研究与发展
30. GPU 实现的高速 FIR 数字滤波算法
31. 一种基于 CPUGPU 异构计算的混合编程模型
32. 面向 OpenCL 模型的 GPU 性能优化
33. 基于 GPU 的 FDTD 算法
34. 基于 GPU 的瑕疵检测
35. 基于 GPU 通用计算的分析与研究
36. 面向 OpenCL 架构的 GPGPU 量化性能模型
37. 基于 OpenCL 的图像积分图算法优化研究
38. 基于 OpenCL 的均值平移算法在多个众核平台的性能优化研究
39. 基于 OpenCL 的异构系统并行编程
40. 嵌入式系统中热备份双机切换技术研究
41. EFI-Tiano 环境下的 AES 算法应用模型
42. EFI 及其安全性研究
43. 基于 UEFI Shell 的 PreOS Application 的开发与研究
44. UEFI Bootkit 模型与分析
45. UEFI 计算机系统快速调试方法的实现
46. 基于 EFI 系统的多文件系统解决方案
47. 基于 UEFI 的可信 Tiano 设计与研究

Programming:

1. 计算机软件基础数据结构 - 算法
2. 高级数据结构对算法的优化
3. 零基础学算法
4. Linux 环境下基于 TCP 的 Socket 编程浅析
5. Linux 环境下基于 UDP 的 socket 编程浅析
6. 基于 Socket 的网络编程技术及其实现
7. 数据结构考题 - 第 1 章 绪论
8. 数据结构考题 - 第 2 章 线性表
9. 数据结构考题 - 第 2 章 线性表 - 答案
10. 基于小波变换与偏微分方程的图像分解及边缘检测

11. 基于图像能量的布匹瑕疵检测方法
12. 基于 OpenCL 的拉普拉斯图像增强算法优化研究
13. 异构平台上基于 OpenCL 的 FFT 实现与优化
14. 数据结构考题 - 第 4 章 串
15. 数据结构考题 - 第 4 章 串答案
16. 用 IPv6 编程接口实现有连接通信的方法
17. 一种战棋游戏的 AI 算法设计与实现浅析
18. 基于 TLS 协议的 ECC 扩展研究

FPGA / CPLD:

1. 一种基于并行处理器的快速车道线检测系统及 FPGA 实现
2. 基于 FPGA 和 DSP 的 DBF 实现
3. 高速浮点运算单元的 FPGA 实现
4. DLMS 算法的脉动阵结构设计及 FPGA 实现
5. 一种基于 FPGA 的 3DES 加密算法实现
6. 可编程 FIR 滤波器的 FPGA 实现
7. 基于 FPGA 的 AES 加密算法的高速实现
8. 基于 FPGA 的精确时钟同步方法
9. 应用分布式算法在 FPGA 平台实现 FIR 低通滤波器
10. 流水线技术在用 FPGA 实现高速 DSP 运算中的应用
11. 基于 FPGA 的 CAN 总线通信节点设计
12. 基于 FPGA 的高速时钟数据恢复电路的实现
13. 基于 FPGA 的高阶高速 FIR 滤波器设计与实现
14. 基于 FPGA 高效实现 FIR 滤波器的研究
15. FPGA 的 VHDL 设计策略
16. 用 FPGA 实现串口通信的设计
17. GPIB 接口的 FPGA 实现
18. 一种基于 FPGA 的 FFT 阵列处理器
19. 基于 FPGA 的 FFT 信号处理器的硬件实现
20. CPLD 在 CAN 通讯卡中的应用
21. 用 CPLD 实现同步串口与异步串口的转换
22. CPLD 在 LED 网络控制器中的应用
23. 基于 CPLD 的双口 RAM 设计与应用
24. CPLD 在有源电力滤波器中的应用