文章编号: 1001 - 2486(2003)04 - 0053 - 06

面向 I/O 优化调度器的磁盘特征提取*

张 巨, 肖 予钦, 景 宁, 陈 宏盛
(国防科技大学电子科学与工程学院, 湖南 长沙 410073)

摘 要: 外存I/O一直是影响I/O密集型应用系统性能的决定性因素。在系统分析I/O优化调度策略及其对磁盘特征参数的需求的基础上, 提出了一组面向I/O优化调度器的磁盘特征抽取方法。这些抽取方法已经被证明是高效的, 并且在 Traxtents-Cello 调度器的实现中得到采用。

关键词: 磁盘驱动器; SCSI; IDE; 磁盘调度; 磁盘特征抽取
中图分类号:TP311 文献标识码: A

The Extraction of Disk Characteristics for I/O Optimized Schedulers

ZHANG Ju, XIAO Yu-qin, JING Ning, CHEN Hong-sheng
(College of Electronic Science and Engineering, National Univ. of Defense Technology, Changsha 410073, China)

Abstract: IOs of the external memory are still the key factor to the performance of IO-intensive applications. Based on the systematical analysis of I/O-optimized scheduling policies and their requirement to characteristics parameters of disk, this paper introduces a set of I/O-scheduler-oriented disk characteristics extracting methods. These methods have been proved efficient and introduced into the implementing of Traxtents-Cello scheduler.

Key words: disk driver; SCSI; IDE; disk scheduling; disk characteristics extraction

由于机械和存储介质技术的发展滞后于芯片制造技术, 外存I/O一直是影响I/O密集型应用系统性能的决定性因素。认识到磁盘吞吐能力的重要性以后, 人们开发了很多用以提高磁盘吞吐能力的I/O调度算法。最初, 这些算法只适用于具体型号的磁盘, 从而给通用的系统软件开发带来了诸多不便。为了解决这个问题, 业界最终达成了旨在磁盘与操作系统间引入一个抽象层以标准化硬盘接口的协议。标准化接口将硬盘的寻址机制逻辑地映射为连续的一维数组, 这就是目前通用的逻辑块号 / 逻辑块地址(LBN/LBA) 寻址模式。磁盘接口的标准化为操作系统设计提供了便利, 但并没有给磁盘性能优化带来任何好处。在系统分析I/O优化调度策略及其对磁盘特征的需求的基础上, 本文提出一组面向I/O调度的磁盘特征抽取方法。这些抽取方法以严格的理论推导为基础, 从而弥补了其它方法在理论上的不足。

1 磁盘特征与基于磁盘特征的 I/O 调度策略

1.1 磁盘特征

磁盘特征分为几何布局、机械参数、盘内 Cache、盘内调度和外部接口等几个方面:

几何布局(Geometry Layouts): 每磁道(Track) 扇区数、每柱面(Cylinder) 磁道数、环带划分(ZBR)、逻辑块地址到物理地址(ZCHS) 的换算、坏扇区替代机制、保留扇区 / 磁道 / 柱面的几何分布、磁道内的扇区组织、磁道 / 柱面 / 环带边界、磁道 / 柱面螺旋偏移(Skew) 等[6]。

机械参数(Mechanical Parameters): 磁盘旋转时间(DRT)、寻道时间、磁头 / 柱面交换时间(H/CST)、写调整时间(WST)、磁头当前位置、扇区旋转时间(SRT)、控制器开销(COT) 等。

盘内 Cache: Cache 管理策略、Cache 段划分(读 / 写段数量、段大小等)、预读策略、Read/Write-On-Arrival 能力等。

* 收稿日期: 2003 - 03 - 05
作者简介: 张巨(1974—), 男, 博士生。

盘内调度: 盘内调度并不是必需的, 如 IDE 硬盘以 PIO 模式提供请求服务时, 每接收到一个请求, IDE 接口即设置为 BUSY 状态, 故不支持盘内命令排队。

外部接口: SCSI/ IDE、数据线宽度、传输速率、支持的命令集等。由于 IDE 硬盘接口封装得过于严格, 大部分高级的磁盘调度算法只能在 SCSI 硬盘上进行, 我们的算法也是面向 SCSI 硬盘的。

为了论述方便, 首先给出几个磁盘参数的意义、表示式以及这些参数间的相互关系。T_{seek}: 磁头寻道时间; $T_{headswitch}$: 磁头交换时间; T_{rotate}: 寻道结束后, 磁头到达请求目标的第一个扇区所经历的时间, 又称为旋转延迟(Rotational Latency); RotateTime: 磁盘旋转一周所需要的时间; SPT: 当前磁道的最大物理扇区数目; size: 前台请求的扇区数; $T_{position}$: 寻道开始后, 磁头到达目标扇区前所经历的时间; $T_{media-access}$: 磁头读写磁道的时间; $T_{service}$: 从寻道开始到磁头读写结束所经历的时间。上述参数的关系如下: $T_{position} = T_{seek} + T_{headswitch} + T_{rotate}$; $T_{service} = T_{position} + T_{media-access}$。如果一次连续读写操作发生在同一磁道上, 那么 $T_{media-access} = RotateTime \times size \div SPT$。

1.2 基于磁盘特征的 I/O 调度策略

LBN/ LBA 寻址模式的引入将磁盘调度自然地划分为盘内调度和前台调度两种类型。盘内调度通常硬编码在磁盘逻辑控制器上; 前台调度则可以通过操作系统的磁盘驱动程序甚至用户空间应用程序予以实现。前台调度的研究主要集中在提高磁盘吞吐量、减少请求响应时间和避免请求"饿死"(指某个或某些请求长时间没有得到执行或响应) 等问题上[3]。我们按照机械指标将磁盘调度策略分为基于寻道时间、定位时间、服务时间和最后期限等四大类, 具体地:

• 基于寻道时间优化的策略

最短寻道时间优先算法(SSTF 或 SSF) 是这类算法的基础。SSTF 在请求队列中选择寻道时间最短的请求作为最优先请求。SSTF 的明显缺陷是位于磁盘最大柱面和最小柱面附近的请求可能会被"饿死", 为解决这种"公平性问题", 又提出了 SCAN、CSCAN 和 FSCAN 等改进算法。SCAN 又称为电梯算法, 它每次仅在一个方向上优先服务最小寻道时间的请求, 当这个方向上没有请求排队时, 再调换方向进行下一轮搜索。SCAN 在一定程度上缓解了公平性问题, 但中间柱面的访问几率仍然比两端要大。CSCAN(Cyclical SCAN) 在单向搜索达到边缘时, 不是调换方向继续搜索, 而是自动寻道到 0 磁道, 然后才进行下一轮搜索。FSCAN(Freezing SCAN) 不是选择寻道时间最小的请求, 而是根据寻道时间进行排序, 一旦排序完成, 则将请求队列冻结起来, 直到这些请求全部完成才处理下一拨请求。

现代操作系统一般采用 LOOK 算法及其变种。比如, CLOOK(Circular LOOK) 按照逻辑地址递增的顺序服务下一个请求, 它可以在没有寻道时间信息的情况下逼真地模拟 SCAN; Linux 操作系统的 ELEVATOR _ LINUS[7] 算法就是 CLOOK 的典型实现。

• 基于定位时间优化的策略

最短定位时间优先算法(SPTF) 是这类算法的基础。SPTF 在请求队列中选择定位时间最短的请求作为最优先请求。这类方法统筹考虑了介质空闲(Media-Free) 时间, 因而在提高磁盘平均吞吐能力方面优于基于寻道时间优化的策略, 但这类算法除需要获得磁盘寻道曲线[9] 外, 还需要精确估算相应请求的旋转延迟。

• 基于服务时间优化的策略

最短服务时间优先算法(STF/ SATF) 是这类算法的基础。STF 在请求队列中选择服务时间最短的请求作为最优先请求。STF 也存在公平性问题, 为解决这一问题, 又提出了 GSTF 和 WSTF 等改进算法。GSTF(Grouped STF) 将请求队列中的请求分成若干组, 在组内采用 STF 算法排队, 只有当前服务组中不再有请求时, 才服务下一请求组; WSTF(Weighted STF) 在预测请求服务时间时, 对请求在队列中的等待时间予以统筹考虑。另外, ASATF(Aged SATF) 的思想与 WSTF 相似, 它最初被应用于 FreeBSD 的进程调

---

* 这里忽略了内外存数据传输等电气参数, 因为这些参数相对于机械参数而言很小。磁头交换时间也是一个电气参数, 但它在旧型号的磁盘中不能忽略。如果数据访问伴随有寻道过程, 那么磁头交换过程通常被淹没在寻道过程中, 故后面的讨论将忽略磁头交换时间。

度。

• 基于最后期限优化的策略

前面三类算法以提高磁盘吞吐量为目标, 而面向实时应用的磁盘调度更关心请求的响应时间。实时调度的核心是面向强制 / 半强制服务最后期限(Deadline) 的优化。其中最简单的是最早最后期限优先(EDF) , 它按照请求所提供的最后期限(Deadline) 的增量顺序进行调度。EDF 没有考虑寻道和旋转延迟时间。为了有效利用磁盘带宽, 提出了许多改进方法, 如: PSCAN(Priority SCAN)、ED-SCAN、FD-SCAN、SCAN-EDF、SSEDO/SSEDV 等等。

2　面向 I/O 优化调度的磁盘特征抽取

2.1　磁盘特征抽取技术回顾

磁盘特征抽取方法可以分为两种类型: 通过接口命令实现的询问(interrogative) 法和通过微基准(Micro Benchmark) 程序实现的测试法。在测试法中, 用于几何参数和机械参数抽取的微基准程序一般采用基于先验知识的统计方法实现; 而用于 Cache 和内部调度特征的抽取则可以采用假设 — 试探法予以实现。

文献[4] 是最早研究 SCSI 磁盘特征抽取方法的成果, 它给出了几个典型参数的询问和经验抽取方法, 其经验抽取方法通过一组同态的 MTBRC 抽取技术实现(MTBRC 的定义如图 1 所示)。MTBRC 方法体现了黑盒(Black Box) 测试方法学理念, 但这种方法抽取某些几何参数(如螺旋偏移) 的时间开销太大而且精度也不是很高。

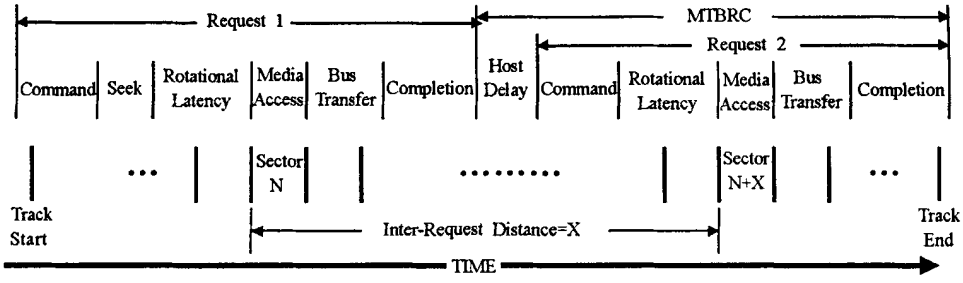


图 1　MTBRC 的定义

Fig. 1　The definition of MTBRC

下面作为示例给出通过 MTBRC 实现磁头交换时间(HST) 的抽取过程:

$MTBRC_1$ = MTBRC(1-sector write, 1-sector read on the same track);

$MTBRC_2$ = MTBRC(1-sector write, 1-sector read on a different track of the same cylinder);

HST = ($MTBRC_2$ − $MTBRC_1$) − ($HostDelay_2$ − $HostDelay_1$)

文献[5] 提出的 DIXtrac 方法通过预先提供的各种启发式规则, 能够自动抽取磁盘特征参数。文献[1] 提出了一个基于 Linux SCSI Generic 实现的 SCSIBench 原型, 我们的抽取算法就是以 SCSIBench 为原型实现的。文献[8] 提出了一套更加细致的特征抽取方法, 其作者声称其方法也适用于 IDE 磁盘, 但其抽取方法需要很多先验的磁盘特征知识。文献[2] 提出了一种能够抽取大部分几何参数的 SKIPPY 基准程序, 为了弥补 SKIPPY 只能抽取磁盘局部特征的不足, 其作者又给出了分别抽取区带信息和寻道曲线的 ZONED 和 SEEKER 微基准程序。文献[2] 的最大特点在于它采用读写原始设备(Raw Device) 接口的方法实现了上述三个微基准程序, 从而保证其对 SCSI 和 IDE 都是适用的, 但其抽取精度很低。

2.2　面向 I/O 优化调度器的磁盘特征抽取理论

寻道时间和旋转延迟的计算是前面提到的四类调度策略的前提。我们所指的面向 I/O 优化调度的磁盘特征抽取实际上就是针对寻道时间和旋转延迟的。

• 寻道时间 T_{seek}

T_{seek} 是由磁头起始柱面号、目标柱面号和寻道方向所决定的经验函数。一般地, T_{seek} 可以表示为下面的形式: $T_{seek}(C_s, d) = f(C_s, C_s + d)$, 其中, $C_s =$ 起始柱面号, $d =$ 起始柱面号 $-$ 目标柱面号。经验表明[4,5], C_s 与寻道方向对 T_{seek} 的贡献相对于 d 而言非常小, 故实际上通常采用公式 $T_{seek}(d) = Max\{T_{seek}(C_s, \pm d) \mid For\text{-}Each\ C_s\}$ 拟合寻道曲线, 取 Max{} 的目的是避免因为低估寻道时间而造成磁盘空转。磁头寻道的机械过程为: 加速前进 → 匀速前进 → 减速前进 → 定位调整。当磁头起始柱面与目标柱面的距离非常小时, 磁头的定位调整时间起决定作用; 磁头起始柱面与目标柱面的距离相对较大时, 磁头的加 / 减速时间起决定作用; 当磁头起始柱面与目标柱面的距离很大时, 磁盘的匀速运动时间就起到决定作用了。故有下面的经验公式:

$$T_{seek}(d) = \begin{cases} \text{离散数组} & d \leqslant \text{门限 } 1 \\ c1 + c2 \times d^{1/2} & \text{门限 } 1 < d \leqslant \text{门限 } 2 \\ c3 + c4 \times d & d > \text{门限 } 2 \end{cases} \quad (1)$$

• 磁头定位时间 $T_{position}$

前面已经提到, $T_{position} = T_{seek} + T_{headswitch} + T_{rotate}$, 忽略 $T_{headswitch}$ 后, 惟一的未决因素就是 T_{rotate} 了。定义 S_1 代表目标扇区, S_0 代表当前磁头所在扇区, S_{rotate} 代表到达目标柱面但未到达目标扇区前磁头所掠过的扇区数量, 则根据文献[3] 的推导, 有:

$$T_{rotate} = S_{rotate} \times RotateTime / SPT$$

$$S_{rotate} = (S_1 + \sum Skew_i - T_{seek} \times SPT / RotateTime - S_0) \bmod SPT \quad (2)$$

其中, $\sum Skew_i$ 表示 S_1 与 S_0 之间所有磁道的螺旋偏移总和。

式(2) 仅仅在起始柱面与目标柱面的 SPT 相同的情况下才有意义, 对于支持 ZBR 的磁盘, 跨环带的换算公式要考虑起始柱面与目标柱面的 SPT 差异。认真分析后, 我们得出:

$$T_{rotate} = S_{rotate} \times RotateTime / SPT_1$$

$$\begin{aligned} S_{rotate} &= [(S_1 - S_{1-0}) + \sum(Skew_i \times SPT_1 / SPT_i) - T_{seek} \times SPT_1 / RotateTime \\ &\quad - (S_0 - S_{0-0}) \times SPT_1 / SPT_0] \bmod SPT_1 \\ &= SPT_1 \times [(S_1 - S_{1-0}) / SPT_1 + \sum(Skew_i / SPT_i) - T_{seek} / RotateTime \\ &\quad - (S_0 - S_{0-0}) / SPT_0] \bmod SPT_1 \end{aligned} \quad (3)$$

其中, SPT_1 和 SPT_0 分别表示起始柱面与目标柱面的 SPT 值; S_{1-0} 表示 S_1 所在磁道的起始扇区号, S_{0-0} 表示 S_0 所在磁道的起始扇区号。考虑到坏扇区的影响, 式(3) 可能需要根据得到的 $S_1 \sim S_{1-0}$ 和 $S_0 \sim S_{0-0}$ 之间坏扇区数目进行微调。

2.3 磁盘特征抽取的步骤

磁盘特征抽取方法可分为 4 个步骤: (1) 磁盘几何布局抽取; (2) 机械参数抽取; (3) Cache 管理策略抽取; (4) 命令处理与数据传输开销指标抽取。我们实际上只关心前面两类参数的抽取工作。

几何布局的抽取: 几何布局参数抽取的关键是获得 LBN ←→ ZCHS 映射关系, 它可以通过询问或经验抽取方法实现。SCSI 接口提供了通过 SEND/ RECEIVE DIAGNOSTIC 的 Translate 选项实现 LBN ←→ ZCHS 互转换的询问方法, 并提供用以定位所有坏扇区的 READ DEFECT LIST 命令作为地址翻译命令的补充。通过 SEND/ RECEIVE DIAGNOSTIC 查询每一个扇区的 LBN ←→ ZCHS 映射关系显然是耗时且没有必要的。在没有坏扇区影响的情况下, 只要获得两个连续磁道的起始扇区 LBN, 就可以换算其中扇区号较小的磁道的全部映射关系。有了 LBN ←→ ZCHS 映射关系和 DEFECT LIST, 就可以获得磁盘的扇区每磁道、磁道每柱面、ZBR、坏扇区的替代机制、保留扇区 / 磁道 / 柱面的几何布局、磁道内扇区的组织、磁道 / 柱面 / 环带边界等特征信息。螺旋偏移(Skew) 的抽取要复杂一些, 下面给出一个简单的 Skew 抽取方法, 在后面的论述中进一步探讨螺旋偏移的抽取思路:

STEP1: 通过 MODE SELECT 命令关 Write Cache 或在命令中置 FUA(强制单元存取) 位以屏蔽 Cache 的影响;

STEP2: 在两个逻辑上连续但位于不同磁道 / 柱面的扇区执行连续写操作, 记录第二次写操作(Write2) 的响应时间;

STEP3: Skew = $Write_2 \times SPT / RotateTime$

机械参数的抽取: 机械参数的抽取建立在几何布局抽取结果基础之上。其中:

(1) 磁盘旋转时间(DRT) 可以通过 MODE SENSE 命令的几何模式页(Geometry Mode Page) 获得, 或者 DRT = MTBRC(1-sector write, same-sector write)。

(2) 磁头 / 柱面交换时间的基本思想前面已经介绍过。

(3) 写调整时间 WST = MTBRC(1-sector write, 1-sector write on a different track of the same cylinder) - MTBRC(1-sector write, 1-sector write on the same track) - HST。

(4) 磁头当前位置可以通过上一个读 / 写命令的位置和完成时间确定。这种方法虽然会受到 Cache 的影响, 但通过关闭 Cache 或置 FUA 位可以避免其影响; 即使在一般条件下, 因击中 Cache 而造成的影响也不会传播到下一条命令以后的读写命令。

(5) 磁道上的存储介质除了用于存储用户数据的扇区外, 还有为伺服器提供信息的磁盘元数据表(Metadata Table)。但一般 SRT 都用 RotateTime/SPT 近似表示, 控制器开销也忽略不计。

(6) 寻道曲线的抽取依赖于应用的目的, 磁盘建模时通常使用平均寻道时间曲线, 而磁盘调度算法通常采用更保守的最大寻道时间曲线。文献[4] 提出了一种长程 SEEK 与短程 MTBRC 相结合的高速最大/ 平均寻道曲线抽取方法, 但这种方法并不实用, 因为很多型号的 SCSI 硬盘在执行 SEEK 命令时仅仅是将磁头放置到目标柱面附近或者什么都不做就返回。我们用文献[1] 中提出的变步长 MTBRC 方法解决了寻道曲线的高速抽取问题[9]。

式(3) 给出了 T_{rotate} 理论上精确的计算公式, 但这个公式并不实用, 因为其计算开销与柱面间距离成正比且会受到累积误差的影响。为了解决计算量问题, 可以预先处理计算开销大且一成不变的螺旋偏移求和计算并将其保存到磁盘文件中, 以后的计算就可以通过读取该文件的相应数据项实现了。具体方法如下:

选择磁盘的(C_0, H_0, S_0) 作为参考点;

FOR (每一个磁道 i)

{

$$TS_I = \sum_{j=0}^{i} (Skew_j / SPT_j);$$

将 TS_I 存储到磁道 i 的特征信息表中;

}// 这里只是为了表达直观, 实际的计算过程可以通过增量计算的方法实现

这样, 每次运用式(3) 时, $\sum(Skew_i / SPT_i)$ 项就可以用 $|TS_I - TS_0|$ 替代。为了规避累积误差的影响, 可以利用测量并存储[(MTBRC$_{wr}$($C_0H_0S_0$, 目标磁道的起始扇区) - MTBRC$_{wr}$($C_0H_0S_0$, 目标磁道))/RotateTime] 以取代 $\sum_{j=0}^{i}(Skew_j / SPT_j)$ 的测算。

3　参数抽取结果及结论

我们的实验环境为: 2.4.8 内核的 Mandrake Linux 8.1 操作系统、Seagate 18GB 单碟10 000 转 SCSI-3 硬盘、Adaptec 80MBps SCSI-PCI 适配卡、133MBps PCI-X 系统总线。

参数抽取程序中采用在 C++ 中嵌入 RDTSC(Pentium 系列 CPU 专用) 汇编指令读取 CPU 时钟的方法代替了基于 Proc 文件的系统时钟, 从而保证测试程序能够获得 ns 级精度的墙上时钟。利用可以旁路内存和文件系统管理的 SCSI Generic 接口实现了特征抽取程序。下面给出几组典型的特征抽取结果(见表1、表2、图2)。

表 1　Seagate ST318406 硬盘基本参数

Tab. 1　Basic parameters of Seagate ST318406 disk

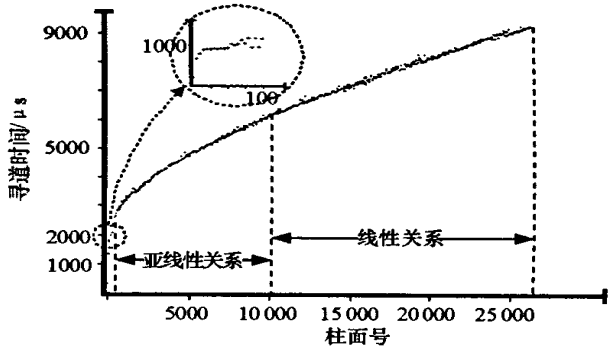
容量(GB)	扇区数	柱面数	磁头数	磁盘转速(RPM)
18.352	35 843 670	26 302	2	测量值10 056,出厂标称10 028

表 2 **ST318406** 的几何布局参数

Tab. 2 Geometric layouts of ST318406

环带	柱面起止	扇区起止	每磁道扇区数	柱面螺旋偏移 μ_s	磁道螺旋偏移 μ_s
1	0– 3741	0– 5 521 710	738	1100.589355	1087.00610
	3761– 6983	5 523 192– 10 278 864			
	7000– 10632	10 280 340– 15 641 171			
	10 651– 12 440	15 642 648– 18 283 205			
2	12 450– 14 223	18 284 688– 20 791 710	707	1164.298706	1100.806152
3	14 233– 17 392	20 793 124– 25 064 091	676	1062.534180	1055.352539
4	17 409– 19 075	25 065 444– 27 251 236	656	1141.457031	1020.420410
5	19 084– 20 466	27 252 548– 29 018 744	639	1114.119141	1040.964844
6	20 474– 23 052	29 020 022– 32 190 962	615	1109.750977	955.557617
7	23 066– 24 540	32 192 192– 33 931 510	590	1105.793457	1033.001953
8	24 548– 26 292	33 932 692– 35 934 803	574	1178.961914	1050.728516

我们提出的磁盘特征抽取方法以 I/O 优化调度为背景,并且以严格的理论推导为前提。本文给出的方法已经被证明是有效的,并且在 Traxtents-Cello 调度器的实现中得到采用。



$$T_{seek}=\begin{cases}\text{离散数组} & d<50\\ 46.398\times d^{1/2}+1493.5 & 10\,000>d\geqslant 50\ \text{时}\\ 0.198\times d+4293.0 & d\geqslant 10\,000\ \text{时}\end{cases}$$

图 2 ST318406 的寻道曲线

Fig. 2 Seek-curve of ST318406

参 考 文 献:

[1] Dimitrijevic Z, et al. Diskbench: User-level Disk Feature Extraction Tool[C]. ACM FAST, 2002.

[2] Talagala N, et al. Microbenchmark-based Extraction of Local and Global Disk Characteristics[R]. UC Berkeley Technical Report, 1999.

[3] Huang L, et al. Implementation of a Rotation-latency-sensitive Disk Scheduler[R]. SUNY at Stony Brook Technical Report, 2000.

[4] Worthington B L, et al. Online Extraction of SCSI Disk Drive Parameters[C]. ACM SIGMETRICS, 1995.

[5] Schindler J, et al. Automated Disk Drive Characterization[R]. CMU-CS-99-176, 1999.

[6] Schmidt F. The SCSI Bus & IDE Interface Protocols, Applications & Programming[M]. 北京: 中国电力出版社, 2001.

[7] 毛德操,胡希明. LINUX 内核源代码情景分析[M]. 杭州: 浙江大学出版社, 2001.

[8] Aboutabl M, et al. Temporally Determinate Disk Access: An Experimental Approach[R]. Univ. of Maryland Technical Report CS-TR-3752, 1997.

[9] 张巨,等. Traxtents-Cello: 一种新的磁盘调度算法[J]. 计算机学报, 2003.