**廉价冗余磁盘阵列的研究**

***戴维 A.帕特森、加斯·吉布森、兰迪·H·卡茨***

*电气工程与计算机科学系*

*571埃文斯大厅*

*加利福尼亚大学*

*伯克利分校，CA94720*

*（pattrsn@ginger.berkeley.edu）*

*摘要：如果I/O没有相近的性能提高来匹配，CPU和内存的性能提高将会被浪费。尽管单个大型昂贵磁盘（SLED）的容量已经在迅速增长， SLED的性能改进却不大。基于为个人计算机开发的磁盘技术的廉价磁盘冗余阵列（RAID）为SLED提供了一种有吸引力的替代方案，有希望在性能、可靠性、功耗和可扩展性上会有一个数量级的改进。本文将介绍五个级别的RAID，并将RAID与IBM 3380和富士通的Super Eagle进行比较，同时给出它们的相对性价比。*

**1、背景：不断提高的CPU和内存性能**

当前时代，用户们正在享用计算机运行速度空前增长带来的便利。戈登·贝尔表示1974年到1984年间，单芯片计算机的性能每年提高40%，大约是小型计算机的两倍[Bell84]。比尔·乔伊预计在接下来的一年里增长会更快[Joy 85]：

 大型机和超级计算机制造商，难以与“Joy法则”预测的快速增长保持同步，故通过提供多处理器作为他们的顶级产品。

但是目前还没有一个高效的系统可以用来制造一个高效的CPU。阿姆达尔通过使用此规则将CPU速度与主存储器的大小相关联[Siewiorek 82]：

每一个CPU指令每秒需要一个字节的主存储器的存储空间；如果计算机系统的成本不受内存成本的支配，那么阿姆达尔的定律表明存储芯片容量应该以同样的速度增长。戈登·摩尔预测在20年前就预测了增长率：

正如摩尔定律所预测的那样，主存的容量每2年[摩尔75岁]到3年[迈尔斯86岁]翻四倍。

最近，主存的兆字节与MIPS的比率被定义为alpha[Garcia 84]，其中Amdahl的定律表示alpha=1。部分原因是因为内存价格的暴跌，主存大小比CPU速度增长更快，许多机器今天都达到了alpha3或更高的级别。

为了维持计算机系统的成本平衡，二级存储必须与系统其他部分的进展相匹配。衡量磁盘技术的一个关键指标是每平方英寸可存储的最大比特数目的提高，或者磁道中每英寸的比特数乘以每英寸的磁道数的提高。对于最大面积密度，被称为M.A.D的“磁盘密度第一定律”预测[Frank87]：

磁碟技术每三年将容量翻一番，价格减半，这与半导体存储器的增长率一致。并且，在1967年到1979年的实践中，IBM数据处理系统的平均磁盘容量超过了其主存储器的容量。

内存容量并不是为了保持系统平衡而必须快速增长的唯一特性，因为指令和数据传送到CPU的速度也决定了CPU的最终性能。主存储器的速度保持同步有两个原因：

1. 高速缓存的发明，它可以自动管理一个小型缓冲区以包含相当一部分的内存调用。
2. 用于构建缓存的SRAM技术，其速度以每年40%到100%的比例提高。

与主存储器技术相比，单个大型昂贵磁盘（SLED）的性能提高有限。这些机械设备主要由寻道和旋转延迟技术控制：自1971至1981年，高端的IBM磁盘的原始寻道时间仅提高了两倍，而旋转时间没有改变[Harker81]。密度越大，意味着信息被发现时传输率越高，额外的磁头可以减少平均寻道时间，但原始寻道时间仅以每年7%的速度提高。我们没有理由期望在不久的将来会有更快的速度。

为了保持平衡，计算机系统一直在使用更大的主存储器或固态磁盘来缓冲一些输入输出(I/O)行为。对于I/O行为具有参考位置且不具有波动性的应用程序，这可能是一个很好的解决方案，但是，以对小数据的高随机请求率（如事务处理）或对大量数据的低请求率（如在超级计算机上运行的大型模拟）为主的应用程序面临着严重的性能限制。

1. **尚未解决的I/O危机**

在让其他方面不受影响的同时，提高某些问题的性能会带来怎样的变化？安达尔的回答现在也被称为安达尔定律[安达尔67]：

*其中：*

*S=有效加速；*

*f=快速模式下的工作分数；*

*k=在快速模式下的加速。*

假设当前的一些应用程序在I/O上花费了10%的时间，那么当计算机的速度提高10倍时——据Bill Joy在三年前所说——那么Amdahl定律预测，（的）有效的加速率（将会）只有5倍。当计算机的速度提高100倍时——通过单处理器或多处理器的革新——这个应用程序的加速将小于10倍，浪费了90%的潜在加速。

尽管我们可以通过缓冲短期I/O需求对软件文件系统进行改进，但是依旧需要创新来避免I/O危机[Boral83]。

**3、一种解决方案：廉价磁盘阵列**

自从个人计算机为廉价磁盘创造了市场后，快速地提高大型磁盘容量已经不再是磁盘设计者的唯一目标。廉价磁盘具有更低的性能和更低的容量。下面的表1比较了顶级的IBM 3380型AK4大型机磁盘、富士通M2361A“超级鹰”微型计算机磁盘和Conner外围设备CP 3100个人计算机磁盘的多种特征。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *特征* | *IBM*  *3380* | *富士通*  *M2361A* | *Conner*  *CP3100* | *3380对比*  *3100* | *2361对比*  *3100* |
|  |  |  |  | *（大于1意味着3100更好）* | |
| 磁盘直径（英寸） | 14 | 10.5 | 3.5 | 4 | 3 |
| 格式化数据容量（MB） | 7500 | 600 | 100 | 0.01 | 0.2 |
| 价格/MB（包括控制器） | $10-$18 | $17-$20 | $7-$10 | 0.01 | 0.2 |
| 额定的MTTF（小时） | 30,000 | 20,000 | 30,000 | 1 | 1.5 |
| 实际的MTTF（小时） | 100,000 | ? | ? | ? | ? |
| 执行器数目 | 4 | 1 | 1 | 0.2 | 1 |
| 最大I/O数 /秒/执行器 | 50 | 40 | 30 | 0.6 | 0.8 |
| 典型I/O数 /秒/执行器 | 30 | 24 | 20 | 0.7 | 0.8 |
| 最大I/O数 /秒/盒 | 200 | 40 | 30 | 0.2 | 0.8 |
| 典型I/O数 /秒/盒 | 120 | 24 | 20 | 0.2 | 0.8 |
| 传输速率 (MB/秒) | 3 | 2.5 | 1 | 0.3 | 0.4 |
| 功率/盒 (W) | 6,600 | 640 | 10 | 660 | 64 |
| 体积(立方英尺) | 24 | 3.4 | 0.03 | 800 | 110 |

***表1：****用于大型计算机的IBM 3380磁盘模型AK4、用于微型计算机的富士通M2361A“超级鹰”磁盘和用于个人计算机的Conner外围设备CP 3100磁盘的比较。我们所说的“最大I/O/秒”是指单个扇区访问的最大平均寻道次数和平均轮询次数。3380的成本和可靠性信息来自广泛经验[IBM 87] [Gawlick87]，富士通信息的来自手册[富士通87]，而新CP3100的一些数据是基于猜测。因为存在批量购买打折的情况和供应商涨价的情况，所以每兆字节的价格是一个范围。(考虑到外部电源的效率低的情况，CP3100的最大功率由8瓦提高到10瓦，而其他驱动器包含自己的电源。)*

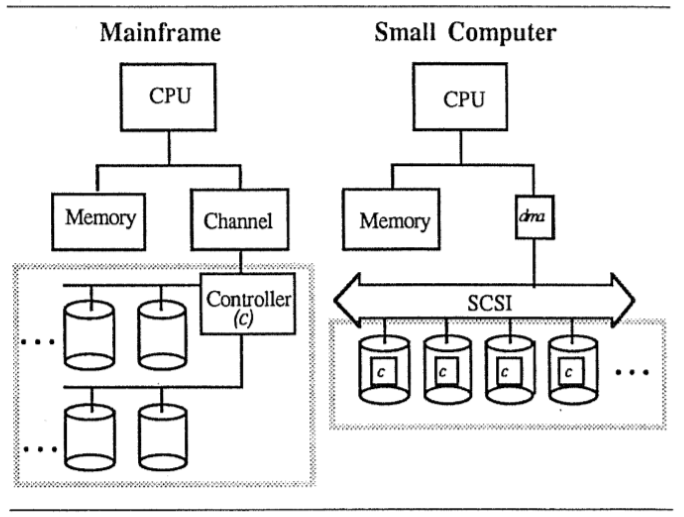
一个令人惊讶的事实是，在一个廉价的磁盘中，每个执行器的每秒读写次数在大型磁盘的1/2以上。在其他几个指标中，包括每兆字节的价格，廉价磁盘等于或优于大磁盘。

自从cp3100这样包含完整的磁道缓冲器和传统大型机控制器大部分功能的磁盘出现以来，人们就更加关注廉价磁盘的小尺寸和低功耗。标准委员会努力消除更高级别的外围接口使得小型磁盘制造商得以在大容量磁盘中提供这些功能，比如ANSI X3.131-1986小型计算机系统接口(SCSI)。这些标准鼓励Adeptec等公司将SCSI接口作为单个芯片提供，从而允许磁盘公司低成本地将大型机控制器功能嵌入。表1比较了传统的大型机磁盘方法和小型计算机磁盘方法。在每个磁盘中作为控制器嵌入的同一SCSI接口芯片也可以用作SCSI总线另一端的直接内存访问(DMA)设备。

基于这些特性，我们建议将I/O系统构建为廉价磁盘阵列，这些磁盘要么交错用于超级计算机的大型传输[Kim 86][Livny 87][Salem86]，要么独立用于许多事务处理的小型传输。由表I中的信息可以看出，75个廉价磁盘有着IBM 3380的12倍的I/O带宽和相同的容量，也具有更低的功耗和成本。

**4、警告**

我们无法在本文中探究与此类阵列相关的所有问题，因此我们将重点放在对性价比和可靠性的基本估计上。我们给出的理由是：如果在性价比方面没有优势，或者在可靠性方面存在严重缺陷，那么就没有必要进一步探索。我们将事务处理的工作负载特征化，以评估廉价磁盘集合的性能，但是请记住，这只是完整的事务处理系统的一个硬件组成。虽然基于这些想法设计一个完整的TPS是很诱人的，但在本文中我们将抵制这种诱惑不予讨论。布线和封装，当然属于许多廉价磁盘阵列的成本和可靠性问题，但也不在本文的讨论范围中。



***图1：****典型大型机和小型计算机磁盘接口的组织比较。单片SCSI接口，如Adaptec A/C-6250，允许小型计算机使用单片作为DMA接口，并为每个磁盘提供嵌入式控制器[Adeptec 87]。(表一中的每兆字节的价格包括了上面阴影框中的所有内容。)*

**5.坏消息：可靠性**

磁盘的不可靠性迫使计算机系统管理人员在发生故障时经常对信息进行备份。如果磁盘增加了100倍，对可靠性有什么影响？假设这些故障都是独立的，并且故障率恒定，故障前时间呈现指数分布，当硬盘制造商在计算平均故障时间(MTIF)时使用以上假设，那么磁盘阵列的可靠性是:

使用表I中的信息，100个CP 3100磁盘的MTTF是30000 /100 = 300小时，少于2周。与IBM 3380的30,000小时(> 3年)的MTTF相比，这是很令人沮丧的。如果我们考虑将阵列扩展到1000个磁盘，那么MTTF是30个小时，大约一天，这需要一个比“令人沮丧”更糟糕的形容词来形容。

如果没有容错机制，大量的廉价磁盘阵列就太不可靠了，无法发挥作用。

**6.更好的解决方案：RAID**

要克服可靠性方面的挑战，我们必须在磁盘出现故障时使用包含冗余信息的额外磁盘来恢复原始信息。廉价磁盘冗余阵列的缩写是RAID。为了简化对我们最终建议的解释，并避免与前面的工作混淆，我们给出了五种不同的磁盘阵列组织的分类，从镜像磁盘开始，通过各种具有不同性能和可靠性的替代方案进行分类。我们将每个组织分类称为RAID级别。

读者应该提前了解，我们描述所有级别时，就好像在硬件中实现只是为了简化表示，事实上，RAID思想不仅适用于硬件，也适用于软件实现。

**可靠性。**我们的基本方法是将阵列分成可靠性组，每个组都有额外的包含冗余信息的“校验”磁盘。当一个磁盘发生故障时，我们假设在短时间内，故障的磁盘将被替换，信息将由冗余信息在新的磁盘中被修复。

这段时间叫做平均修复时间（MTTR）。如果系统有额外的“热”备用的额外磁盘，MTTR时间将会减少；当一个磁盘出现故障时，替换磁盘会通过电子装置连接上去。人工操作人员会定期更换所有故障磁盘。下面是我们使用的其他符号的含义：

*；*

*；*

*；*

*；*

如上所述，我们做相同的假设，即各个磁盘制造商的磁盘故障时间是呈指数分布且之间是相互独立的。（当发生地震或功率激增时，一组磁盘阵列故障时间的分布之间可能不是独立的。）由于可靠性的预测数值将会变得非常高，我们想要强调的是，可靠性仅限于具有此故障模型的磁盘头组件，而不是整个软件和电子系统。另外，在我们看来，技术的进步意味着太高的MTTF值是“过头的”——因为用户终将会替换过时的硬盘，这与预期寿命无关。毕竟，有多少人还在使用20年前的硬盘呢？

计算修复RAID单一错误的大致平均故障时间分为两分步骤。首先，一组磁盘的MTTF为：

其中a代表在某一组磁盘中，某一损坏磁盘修复故障前，组中另一磁盘发生故障的概率。

如附录中更为正式的推导，在第一个磁盘修复前，另一个磁盘发生故障的概率为：

附录中正式推导背后的依据来自于在修复x个单个磁盘故障期间时试图计算的第二个磁盘故障的平均数量。由于我们假设磁盘故障以统一的速率发生，因此在修复X 个单个磁盘故障的期间内这一平均故障数为：

然后是单个磁盘的平均第二次故障次数

其余磁盘的MTTF就是单个磁盘的MTTF除以组中没有故障的磁盘的数量，由此得到了上面的结果。第二步是整个系统的可靠性，大致为(因为不是指数分布):

把他们组合在一起我们得到：

由于在所有的RAID级别中方程式都是一致的，所以我们根据实际需要使用以下数字来具体化抽象参数：总数据磁盘D=100，每一组包含G=10个数据磁盘，MTTFDisk=30000小时，MTTR=1小时，每个组的校验磁盘C的大小由RAID级别决定。

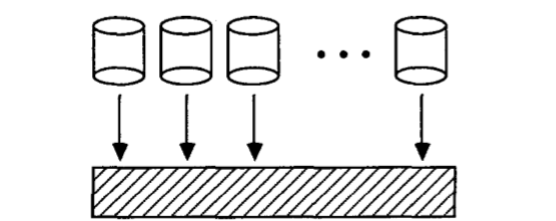
***可靠性系统成本：***这里仅指额外的校验磁盘，表示为与数据磁盘数量D的百分比。正如我们接下来要看到的，开销随RAID级别的变化而变化，变化范围从4%到100%。

***可用存储容量百分比：***另一种表示这种可靠性间接开销的方法是用数据磁盘和可用于存储数据的检查磁盘总容量的百分比表示。根据配置的不同，这一比例从低的50%到高的96%不等。

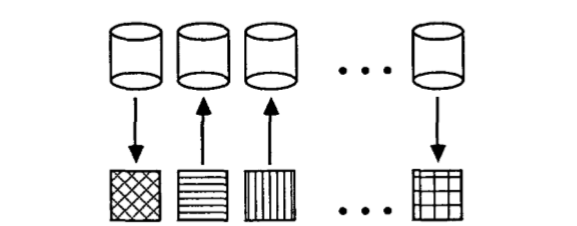
***性能：***由于超级计算机应用程序和事务处理系统具有不同的访问模式和速率，对于这两者，我们需要不同的衡量指标来评估。对于超级计算机，我们统计大型数据块每秒的读写次数，其中“大型”的概念被定义为从一组数据块中的每个数据磁盘至少获取一个扇区的时间。在大数据传输过程中，一组数据块中的所有磁盘充当一个单元，每个磁盘并行读取或写入大数据块的一部分。

一种更好的衡量事务处理系统的标准是每秒的独立读写次数。由于事务处理系统（例如，借方/贷方系统）需要按照读取-修改-写入的顺序访问磁盘，所以我们也包含了这种度量。理想情况下，在小规模传输期间，组中的每个磁盘都可以独立工作，读取或写入独立的信息。总之，超级计算机的应用程序需要使用很高的数据传输速率，而事务处理需要很高的I/O速率。

对于大规模传输和小规模传输的计算，我们假定最小的用户请求单位是一个扇区，一个扇区要比一个磁道小很多，并且有足够的处理任务使每个设备保持运转。因此，扇区大小同时影响了磁盘存储效率和传输数据的大小。图2显示了理想情况下，RAID中大规模磁盘访问和小规模磁盘访问的操作。



(a)单次大规模读取或分组式的读取方式（整个数据磁盘的单次读取）



(b)多次小规模读写或单独的读写方式（多个数据磁盘的多次数据读写）

***图2*** *大规模传输相对于一组数据磁盘中的多次小规模传输。*

这六个性能指标是对于大规模（分组）或小规模（单独）传输的每秒“读取-写入”和“修改-读取-写入”次数。当我们计算效率时，只需要统计一个磁盘阵列每秒的事件数相对于单个磁盘每秒的事件数，而不必给出每个指标的绝对值。（这是Boral的每千兆字节的读/写带宽[Boral 83]通过磁盘扩展到数千兆字节中。）在本文中，我们关注最基本的差异，因此我们使用简单的、确定的吞吐量度量来衡量性能指标，而不是使用延迟来衡量性能指标。

***每个磁盘的有效性能***。磁盘的成本可能占数据库系统成本的很大一部分，因此每个磁盘的读/写性能，包括校验磁盘的开销体现了一个系统的性价比，因而决定了一个廉价冗余磁盘阵列可接受的最低价格。

**RAID1：镜像磁盘**

镜像磁盘是提升磁盘可靠性的一种传统的方式。这是我们所认为最贵的选择因为所有的磁盘都是可复制的（G=1且C=1），而且一个磁盘上的每次写入也是对查核磁盘的一次写入。串联组将错误承受的控制者的数据加倍，允许一个能够使读写相应出现的镜像磁盘的最佳版本。表II显示一个RAID第一层面假设这个最优情况的度量标准。

|  |  |  |
| --- | --- | --- |
| 平均无故障时间  磁盘总数  *系统开销*  可用*存储容量* | 超过产品使用寿命  （4500000小时或>500年）  2D  100%  50% | |
| 事件/秒 vs 单磁盘  *大量（或成组）读*  *大量（或成组）写*  *大量（或成组）读-改-写*  *少量（或单独）读*  *少量（或单独）写*  *少量（或单独）读-改-写* | *整个*RAID  2D/S  D/S  4D/3S  2D  D  4D/3 | 每盘效率  1.00/S  .50/S  .67/S  1.00  .50  .67 |

**表II**.*RAID1的特征。这里我们假定写入操作并没有因为等待二次写入而变慢，因为为了等待2个磁盘的减速与为了写入10到25个磁盘的一整组的等待时间S的减速相比是很微小的。不同于使用一块对软件不可见的额外磁盘的“纯”镜像体制，我们假定一个带有双倍的控制器的最优化体制允许对所有磁盘进行并行读取操作，给大量读取操作以完全的磁盘带宽，并且允许读-改-写过程的读取操作并行出现。*

当单独的访问在大量的磁盘中分散时，平均排队时延、寻道时间和旋转延迟可能因各个磁盘情况而不同。即使带宽可能不变，因为访问分散得更均匀，所以排队时延的变化幅度可能会减小，同时在如果磁盘负载不是很高的情况下，可能出现的排队时延也会因为并行访问而降低[Livny 87]。很多磁盘臂移动到相同的磁道然后旋转到被描述的扇区时，平均的寻道和旋转时间会比单一磁盘的平均时间更长，趋向于最坏的情况。在并行访问多个扇区时这个影响通常不大于对单一扇区的平均访问时间的两倍。在具有足够控制器的某些镜像磁盘的特殊情况中，可以读入任意数据扇区的磁盘臂间的选择会节省最多45%读入时的平均寻道时间[Bitton 88]。

考虑到这些因素，当一个磁盘组里存在两个及两个以上的磁盘时，我们应用了一个减速因子S来保持我们的基本重点。通常来说，每当磁盘组并行工作时，1≤S≤2。对于同步的磁盘，在磁盘组中所有磁盘的主轴都是同步的，以便于与磁盘同组对应匹配的各扇区同时从磁头的下方通过，所以对于同步的磁盘来说没有减速并且S=1。由于RAID1的磁盘组里仅有一个数据磁盘，我们设想在大型的转移中所需要的磁盘个数和在更高级RAID的组中磁盘个数一样，都是10到25个。

复制所有的磁盘意味着仅仅使用50%的磁盘储存容量而整个数据库系统的成本却翻倍。如此多的资源浪费促进了下一级别RAID的诞生。

1. **RAID2：用于错误检测与修正的汉明码**

主存储器结构的发展史启示了一种减少可靠性花费的方法。随着4K和16K 的DRAM的引入，计算机设计师发现有些新型设备因为α粒子而导致信息流失。由于在一个系统中有很多单独的DRAM比特并且在16到64芯片的组中它们通常被同时访问，系统设计师在每个组中增加了冗余的芯片来纠正单一错误和检测双倍同时错误。这使记忆芯片的数量增加了12%到38%（取决于组的大小），但却显著地提高了可靠性。

只要同组所有的数据位同时被读取或写入就不会对性能产生影响。然而，读取的数量小于组的大小时则需要读取整个组以确保信息的正确性，而写入组时需要以下三个步骤：

1. 一个读取步骤来获取剩下的全部数据；
2. 一个修饰步骤来合并新旧数据；
3. 一个写入步骤来把包括检查信息在内的数据写满整个组。

由于在一个RAID中有大量的磁盘，并且有些访问是对磁盘组的，我们可以模仿DRAM的解决方法将同组整个磁盘上的数据位交错再添加足够的检测芯片来探测并纠正单一错误。一个校验磁盘可以探测单一错误，但我们需要足够的检验磁盘来识别出有错误的磁盘来纠正错误。对于一个有十个数据磁盘的磁盘组（G），总共需要4个检验磁盘（C），如果G=25，则C=5。为减少额外部分的成本，我们假定磁盘组的大小从10到25之间变化。

因为我们的单一数据传输单元只是一个部分（扇区），数据位交错的磁盘意味着对RAID的一个巨大传输必须至少为G数量的扇区。如同DRAMs，读取少量数据意味着要从组内每个数据位交错的磁盘中读取全部扇区；单一单元的数据写入包括对所有磁盘的“读取-修改-写入”循环。表三显示了RAID2的度量标准。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 平均无故障时间 | | 超过产品使用寿命 | | | |
| 磁盘总数  系统开销  可用存储容量 | | G=10（494500小时或>50年）  1.40D  40%  71% | | G=25（103500小时或 12年）  1.20D  20%  83% | |
| *事件/秒*  *（vs 单磁盘）*  *大量读*  *大量写*  *大量读*-*改*-*写*  *小量读*  *小量写*  *小量读-改-写* | *完全RAID*  *D/S*  *D/S*  *D/S*  *D/SG*  *D/2SG*  *D/SG* | *每盘效率* | | *每盘效率* | |
| *L2*  *.71/S*  *.71/S*  *.71/S*  *.07/S*  *.04/S*  *.07/S* | *L2/L1*  *71%*  *143%*  *107%*  *6%*  *6%*  *9%* | *L2*  *.86/S*  *.86/S*  *.86/S*  *.03/S*  *.024/S*  *.03/S* | *L2/L1*  *86%*  *172%*  *129%*  *3%*  *3%*  *4%* |

**表III.** *RAID2的特征。L2/L1栏以百分比形式给出了用RAID1表示的RAID2表现（>100%意味着L2更快）。只要传输单元足够大，可以传输一组中所有的数据磁盘，大型的I/O就能占有每个磁盘的全部带宽，并S分割以允许一组内所有磁盘保持完整。RAID1的大型读取更快，因为数据被复制导致在冗余磁盘也可以独立获取到相应数据。小型I/O仍然需要获取一组中的所有磁盘，所以仅有D/G时，小型I/O能依次发生，被S分割后允许一组磁盘完成。RAID2的小型写入与小型读-改-写类似，因为在新数据可被写入每个扇区中的一部分之前，全部扇区必须被读取。*

对于更大的写入工作，即使使用较少的检查磁盘，RAID2也具有与RAID1相同的性能，因此在每个磁盘上，它的性能均优于RAID1。对于小型数据传输，整个系统或每个磁盘的性能都很糟糕。 必须以较小型的传输访问组中的所有磁盘，从而限制了同时访问D/G的最大数量。 由于访问必须等待所有磁盘完成，因此我们还包含了减速因子S。

因此，RAID2对于超级计算机非常理想的，但不合适于事务处理系统，随着组大小的增加，两个应用程序的每个磁盘的性能差异也会增加。 考虑到这一事实，Thinking Machines Incorporated今年宣布了其连接机器超级计算机“ Data Vault”的RAID2，其G = 32和C = 8，其中包括一个热备用磁盘。

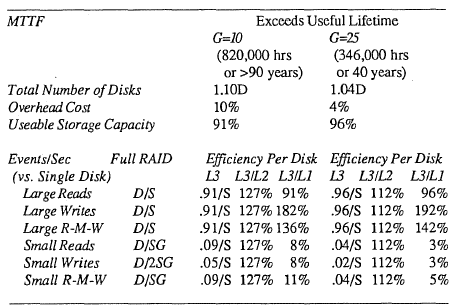
在改善小数据传输之前，我们再次专注于降低成本。

**RAID3：每个组一个检查磁盘**

RAID2中的大多数检查磁盘用于确定哪个磁盘发生了故障，因为只需要一个冗余奇偶校验磁盘即可检测到错误。 这些额外的磁盘是真正的“冗余”，因为大多数磁盘控制器已经可以检测磁盘是否发生故障：通过磁盘接口中提供的特殊信号，或者通过用于检测和纠正软错误的扇区末尾的额外检查信息。 因此，可以通过计算剩余的合格磁盘的奇偶校验，然后逐位比较原始组的奇偶校验来重建故障磁盘上的信息。

当这两个奇偶校验一致时，失败位为0；否则为1。如果检查磁盘发生故障，则只需读取所有数据磁盘并将组奇偶校验存储在替换磁盘中。对于此处考虑的组大小，将检查磁盘减少到每个组一个（C = 1），可以将开销成本降低到4％到10％之间。RAID3系统的性能与RAID2相同，但是每个磁盘的有效性能有所提高，因为它需要较少的检查磁盘。磁盘总数的减少也提高了可靠性，但是由于它仍然大于磁盘的使用寿命，因此这是次要的点。 RAID2相对RAID3的一个优点是不需要与每个扇区关联的额外校验信息来纠正软错误，从而使每个磁盘的容量增加了10％。级别2还允许“即时”纠正所有软错误，而不必重新读取扇区。表IV总结了第三级RAID特性，图3比较了扇区布局并检查了磁盘的2级和3级。

***表****四 RAID3的特征。 L3 / L2列以L2表示L3的％性能，L3 / L1列以L1表示L3的性能（> 100％表示L3更快）。 完整系统的性能在RAID2和RAID 3中相同，但是由于检查磁盘较少，因此每个磁盘的性能得以提高。*



Park和Balasubramanian提出了RAID 3 系统，但未建议特定应用[Park86]。我们的计算表明，它与超级计算机应用程序的匹配比与事务处理系统的匹配要好得多。今年，两家磁盘制造商宣布了使用同步的5.25英寸磁盘（G = 4和C = 1）为此类应用程序提供RAID3：一种来自Maxtor，另一种来自Micropolis [Maginnis 87]。RAID3阵列将可靠性开销成本降低到最低水平，因此在后两级中，我们在不改变成本或可靠性的情况下提高了小型访问的性能。

**10.** RAID4 **：独立读写**

在组内的所有磁盘之间分布传输具有以下优点：

* 因为可以利用整个阵列的传输带宽，所以减少了大型的或成组的传输时间。

但是它也具有以下缺点：

* 读或写组中的磁盘需要读或写组中的所有磁盘； RAID2和RAID3一次只能执行一个I/O操作。
* 如果磁盘不同步，则不会有平均的寻道时间和旋转延迟；观察到的延迟应该趋向于朝最坏的情况移动，因此在上面的方程中就有S因子。

第四层RAID通过并行性提高了小型传输的性能——一次可以在每个组中执行多个I/O操作。我们不再将单个传输信息分布到多个磁盘上，而是将每个单个单元都放在一个磁盘中。

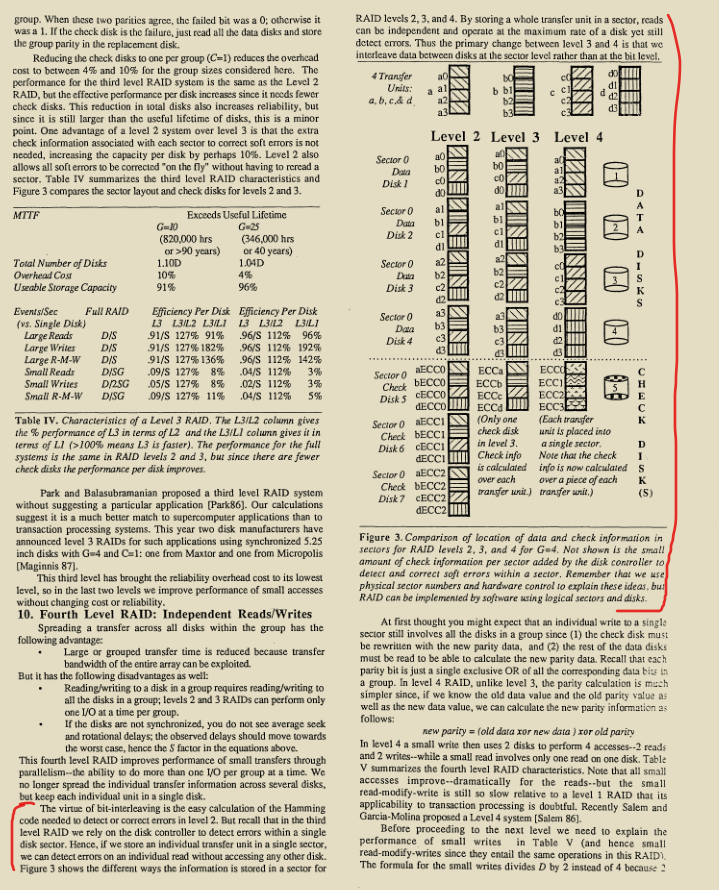
位交错的优点是利于汉明码的计算，以便在第二级磁盘阵列中检测或者纠正错误。但是回想一下，在RAID3中，我们是依赖于磁盘控制器来检测单个磁盘扇区中的错误的。因此，如果把一个独立的传输单元存储在单个扇区中，便可以在不访问任何其他磁盘的情景下检测出单个读取过程出现的错误。如下第三个图形分别展示了RAID2、3、4在扇区中存储信息的不同方式。它将整个传输单元存储在扇区中，使得读取操作变得独立并且在最大速率进行，同时磁盘依旧可以检测到错误。因此，在扇区级别而不是位级别的磁盘之间交错放置数据，是RAID3和RAID4间的最大变化。

*图中的内容：*

*（只有一个检查磁盘在第三级。检查信息是通过每个传输单元进行计算的）。*

*（每个转移单元被放置到一个单独的扇区。注意，现在检查信息在每一块单独的传输单元上进行计算）*。

*图表三（注释）：G=4情况下RAID2、3、4扇区中数据位置和检查信息的比较。此处没有展示出磁盘控制器为检测和纠正每个扇区中的软错误而添加的少量检索信息。请记住，我们使用物理扇区编号和硬件控制来解释磁盘阵列这些道理，但是软件层面它也可以使用逻辑扇区和磁盘来实现。*

****

一开始，你可能认为对单个扇区的单个写操作仍然涉及到组中的所有磁盘，因为：（1）必须用新的奇偶校验数据重写检查磁盘 （2）必须读取其他数据磁盘才能计算新的奇偶校验数据。回想一下，每个奇偶校验位只是一个组中所有对应数据位的单个异或。奇偶校验计算在RAID4中相比于RAID3中要简单的多，如果我们知道旧的数据值、旧的奇偶校验值以及新的数据值，我们可以按如下方式计算新的奇偶校验信息：

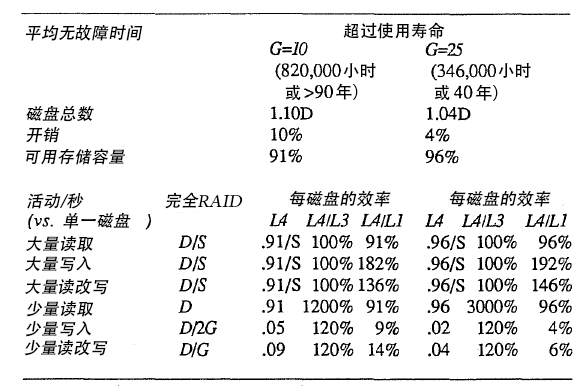
新奇偶校验=（旧数据 异或 新数据）异或 旧奇偶校验

在第4级中，一个小型的写操作使用2个磁盘执行4次访问（2次读取和2次写入），而一个小型的读操作只涉及一个磁盘上的一次读取。表五总结了RAID4特性。需要注意的是，所有的小型访问都得到了改善（对于读操作来说尤为显著），但是相对于一级RAID来说，小型的读写、修改操作仍然非常慢，因此它对事务处理的适用性值得怀疑。最近，Salem和Garcia Molina提出了一个4级系统[Salem 86]。

在进入下一阶段之前，我们需要解释表V中的小型操作性能（由于在这个RAID中需要相同的操作，所以读写、修改操作很小）。小的写操作公式将D除以2而不是4，因为2......

（存储写入的公式把带有数据的磁盘总数D划分成2个磁盘而不是4个磁盘，因为2个磁盘）访问可以并行进行:可以同时读取旧数据和旧奇偶校验，也可以同时写入新数据和新奇偶校验。写入操作的性能还要除以组中的数据磁盘数量G，因为组中的单个检查磁盘必须与该组中的每个写操作一起读写，从而将一次可以执行的写操作数量限制为组的数量。

检验磁盘是一个瓶颈，而最后一个级别的RAID消除了这个瓶颈。



**表格Ⅴ***：RAID4的特征。L4/L3列给出了用L3表示的L4的百分比性能，而L4/L1列给出了用L1表示（如果大于100%就代表L4更快）。读入和读改写稍微提高了，因为我们做了和表格Ⅱ一样的假设：两个相关输入输出流的减速可以忽略，因为只涉及两个磁盘。*

RAID5**：无单校验盘**

尽管RAID4实现了读操作的并行化，但写操作仍然限制在每个组一个，因为每次写操作都必须读写检验磁盘。RAID5将数据和检查信息分布在所有磁盘上——包括检查磁盘。图4比较了RAID4和RAID5中磁盘扇区中的检查信息的位置。

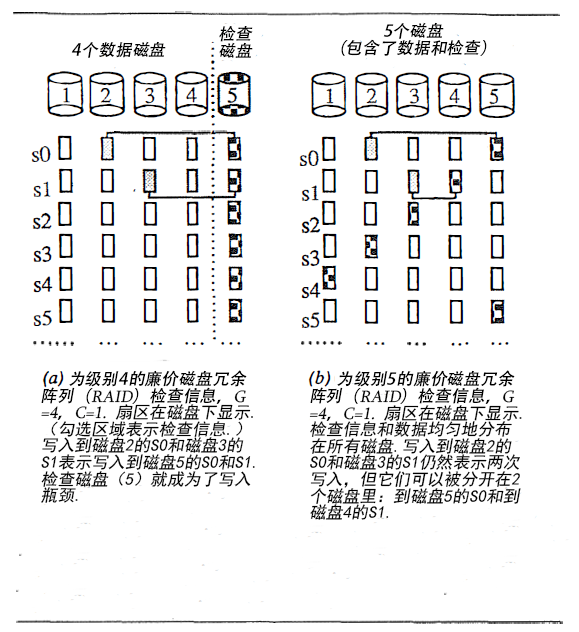
这一小变化对性能的影响很大，因为RAID5支持每个磁盘组中同时进行多个单独的写操作。举个例子，假设在上面的图4中，我们想写入磁盘2的扇区0和磁盘3的扇区1。如图4所示，在RAID4中这些写操作必须是连续的，因为磁盘5的扇区0和扇区1都必须被写。但是，如右图所示，在RAID5中，写操作可以并行进行，因为对磁盘2扇区0的写操作仍然涉及对磁盘5的写操作，而对磁盘3扇区1的写操作涉及对磁盘4的写操作。

这些变化使RAID5接近两个世界最佳水平:小型的读-改-写现在有接近1级RAID的每个磁盘的速度，同时保持RAID3和RAID4的每个磁盘的大型传输性能与高可用存储容量百分比。将数据分散到所有磁盘甚至可以提高小型读取的性能，因为每个组中还有一个磁盘包含数据。表Ⅵ总结了这个RAID的特征。

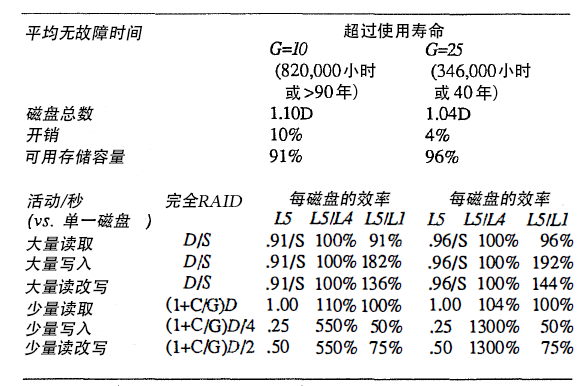
记住先前给出的注意事项，如果你只想做超级计算机应用，或者只是在存储容量有限的时候处理事务，亦或是想同时做超级计算机应用和事务处理， RAID5就显得特别有吸引力。

**12.讨论**

在结束这篇论文之前，我们希望注意有关RAIDs的一些更有趣的地方。首先，虽然磁盘条带和奇偶校验支持的方案是由硬件完成的，但是没有必要这样做。我们只是给出了方法，而硬件和软件解决方案之间的决策是一个严格的成本和效益。例如，在磁盘缓冲有效的情况下，由于旧的数据和奇偶校验将在主内存中，所以对于第5级的小型写入，不需要进行额外的磁盘读取，因此软件将提供最佳的性能和最低的成本。



**图4.** *4级的廉价磁盘冗余阵列（RAID）与5级的廉价磁盘冗余阵列（RAID）每块扇区中校验信息的位置比较*



***表格VI****：5级的廉价磁盘冗余阵列（RAID）的特征. L5/L4列给出了用L4表示L5的百分比性能，而L5/L1则给出了用L1表示的百分比性能（如果大于100%意味着L5更快）。因为读操作可以扩展到包括级别4的校验磁盘在内的所有磁盘，所以所有小型I/O接口都可以提高1 +C/G。因为不再受组大小的限制，可以获得与访问相关的4个I/O的完整磁盘带宽，小型写操作和R-M-W都得到了改进。我们再次做出与表II和表V相同的假设:因为只涉及两个磁盘，我们可以忽略两个相关I/O的减速。*

在本文中，我们假设转移单元是扇形的倍数。随着最小传输单元的大小在每个驱动器上超过一个扇区(例如使用I/O协议支持调优无序数据的完整磁道)，由于每个磁盘中的完整磁道缓冲区，RAID的性能将显著提高。例如，如果每个磁盘一到达下一个扇区就开始传输到它的缓冲区，那么S可能会减少到小于1，因为几乎没有旋转延迟. 对于轨道大小的传输单元，甚至不清楚在一个组中同步磁盘是否可以提高RAID性能。

本文提出了两个可分离观点：

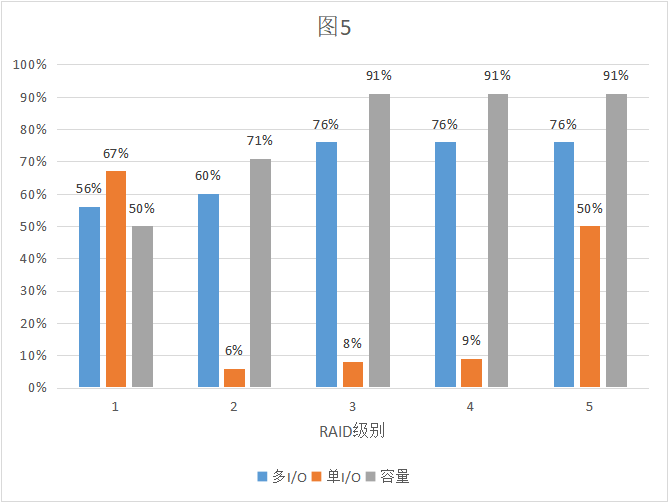
1. 从个人计算机磁盘构建I/O系统的优点和五种不同磁盘阵列组织的优点，都独立于在各自阵列使用的磁盘。
2. 使用传统的在磁盘中创建镜像的方法实现可以接受的可靠性，并且提高了后面这些东西的水平：

* 数据速率，含义是每秒对大量连续信息的少量请求(超级计算机应用程序)；
* 输入输出速率，其特征是对少量随机信息进行大量的读-修改-写操作（事务处理）；
* 或可用的存储容量；
* 或者可能是三个都有。
* 图5显示了每个级别的磁盘阵列中每一磁碟的性能改进。每个磁盘的最高性能来自级别1或5级。在事务处理情况下，使用不超过50%的存储容量，则选择镜像磁盘（级别1）。但是，如果此时要求使用超过50%的存储容量，或者要求用于超级计算机应用程序或用于组合的超级计算机应用程序和事务处理，此时第5级似乎是最好的。复制的数据提高了读取性能，但降低了容量和写入性能，而校验数据只有在出现故障时才有用，因此级别1的优点或者缺点是它复制数据而不是计算校验信息。
* 受到寻呼研究的时间空间积的启发[Denning 78]，我们提出一个称为空间速度积的单个品质因数：可用的存储分数乘以每个事件的效率。使用此度量，级别5与级别1相比，在G=10时读取和写入分别有1.7倍和3.3倍的提高。
* 让我们回到第一点，从个人电脑磁盘上看构建输入输出系统的优势。与传统的单个、大尺寸、价格昂贵的磁盘（SLED）相比，廉价磁盘的冗余阵列（RAID)同样的成本下提供了显著的优势。表七比较了在使用100个便宜的5级数据磁盘与一组IBM 3380公司出产的尺寸为10的磁盘。如您所见，与该SLED相比，级别5 的RAID在性能，可靠性和功耗（以及空调成本）方面提高了约10倍，在尺寸上减少了3倍。表七还比较了使用10种廉价数据磁盘组的RAID级别与一组大小为10的富士通M2361A“超级鹰”。在这次比较，RAID在性能、功耗和大小方面提供了大约5倍的改进，在可靠性方面（计算的）提高了两个数量级以上。
* RAID提供了模块化增长的进一步优势。与此IBM磁盘模型中每次增加100,000美元所限制的7,500 MB不同，RAID可以以组大小(1000 MB需要11,000美元)或(如果允许部分组)磁盘大小(100 MB需要1,100美元)进行增长。另一方面，RAID在比SLED小得多的系统中也很有意义。小的增量成本也使得热备盘更加实用，可以进一步降低MTTR，从而提高大型系统的MTTF。例如，一个1000个磁盘级别的RAID（组大小为10）和几个备用磁盘可以计算出超过45年的MTTF。
* 最后的探讨涉及到从1级或5级RAID出发，来设计一个完整的事物处理系统的前景展望。廉价磁盘每兆字节所消耗电量的大大降低，使得系统设计师可以考虑用电池来备份整个磁盘阵列——110个PC磁盘所需功耗不到两个富士通超级鹰（Fujitsu Super Eagles）。另一种方法是在扩展电源发生故障的情况下，使用一些这样的磁盘来保存用电池备份的主存储器中的内容。这些磁盘的较小容量在重构期间也占用了较少的数据库，从而具有更高的可用性。（需要注意的是，在发生故障时，5级将占用组中的所有磁盘，而1级在重构期间只需要单个镜像磁盘，这使1级在可用性方面具有优势。）

**13.** **结论**

面对处理器和内存速度呈指数级高速增长所带来的挑战，RAIDs为我们提供了一个经济且高效的选项。我们相信个人计算机磁盘的尺寸缩减是磁盘阵列成功的关键，正如戈登·贝尔（Gordon Bell）认为微处理器的尺寸缩减是多处理器成功的关键那样[Bell 85]。在这两种情况下，尺寸的缩减都简化了许多组件的互连以及封装和布线。虽然可以使用大型主机处理器来构造大型阵列（或SLED），但是用相同数量的微处理器（或PC驱动器）来构造阵列自然更加容易。正如贝尔（Bell）创造了术语“multi”来区分由微处理器组成的多处理器一般，我们使用“RAID”一词来标识由个人计算机磁盘组成的磁盘阵列。凭借在性价比、可靠性、功耗和模块化增长方面的优势，我们期望RAID在未来的I / 0系统中取代SLED。然而，针对RAIDs的实用性方面，仍然存在一些尚未解决的问题：

* • RAID对于延迟的影响如何？
* • 假设单个磁盘发生非指数故障，对MTTF（平均失效时间）的计算影响如何？
* • 在使用独立故障模型的情况下，RAID的有效期/寿命与计算得出的MTTF（平均失效时间）相比如何？
* • 同步磁盘对4级和5级RAID性能的影响有什么区别？
* • “减速”S的实际表现如何？[Livny 87]
* • 有缺陷的扇区如何影响RAID？
* • 您将如何安排I/O到5级RAIDs使得写入并行性最大化？
* • 在事务处理的过程中，磁盘读取是否会出现访问局部性？
* • 数据是否可以自动重新分配到100至1000个磁盘上以减少争用？
* • 磁盘控制器的设计是否会限制RAID性能？
* • 如何构造100至1000个磁盘并将其物理连接至处理器上？
* • 布线对成本，性能和可靠性有何影响？
* • 为了不限制性能，RAID应连接到CPU的何处？内存总线？I/0总线？缓存？
* • 文件系统是否允许对不同的文件使用不同的分类策略？
* • RAID中固态硬盘和WORMs的作用是什么？
* • “并行访问”磁盘对RAID的影响是什么（并行访问读/写磁头下方的每个表面）？



***图V****：以五种级别100D10G的RAID为例，每个磁盘每秒的多（组合）和单（个体）的“读-修改-写”操作，以及可用的存储容量。如有需要，假设所有级别的S因子值统一为1.3.*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 属性 | RAID 5L  (100,10)  (CP3100) | SLED  (IBM 3380) | RAID  V. SLED  (>1优于RAID) | RAID 5L  (10,10)  (CP3100) | SLED  (Fujitsu M2361) | RAID  V. SLED  (>1优于RAID) |
| 格式化数据容量（MB） | 10000 | 7500 | 1.33 | 1000 | 600 | 1.67 |
| 价格/MB（包含控制器） | $11-$8 | $18-$10 | 2.2-.9 | $11-$8 | $20-$17 | 2.5-1.5 |
| 额定失效前时间（时） | 820000 | 30000 | 27.3 | 8200000 | 20000 | 410 |
| 实际失效前时间（时） | ? | 100000 | ? | ? | ? | ? |
| 执行器个数 | 110 | 4 | 22.5 | 11 | 1 | 11 |
| 最大读写的执行器个数 | 30 | 50 | .6 | 30 | 40 | .8 |
| 最大组合“读-修改-写”/盒 | 1250 | 100 | 12.5 | 125 | 20 | 6.2 |
| 最大个体“读-修改-写”/盒 | 825 | 100 | 8.2 | 83 | 12 | 6.9 |
| 典型读写的执行器个数 | 20 | 30 | .7 | 20 | 24 | .8 |
| 典型组合“读-修改-写”/盒 | 833 | 60 | 13.9 | 83 | 12 | 6.9 |
| 典型个体“读-修改-写”/盒 | 550 | 60 | 9.2 | 55 | 12 | 4.6 |
| 体积/盒（立方英尺） | 10 | 24 | 2.4 | 1 | 3.4 | 3.4 |
| 功率/盒（瓦） | 1100 | 6600 | 6.0 | 110 | 640 | 5.8 |
| 最小扩展大小（MB） | 100-1000 | 7500 | 7.5-75 | 100-1000 | 600 | 0.6-6 |

***表格Ⅶ*** *.使用100个Conners & Associates公司CP 31 OOs 磁盘、组数为10的IBM3380磁盘AK5型号和5级RAID的比较；使用10个廉价数据磁盘、组数为10的 富士通M2361A超级鹰和5级RAID的比较。比较列中大于1的数字表示优于RAID.*

**致谢**

我们希望感谢下列参加了这些想法产生的讨论的人士：Michael Stonebraker, John Ousterhout, Doug Johnson, Ken Lutz, Anapum Bhide, Gaetano Boriello,

Mark Hill, David Wood和在1987年秋天加州大学伯克利分校举办的SPATS研讨会上的学生们。我们也要感谢以下人士，他们对本文的编写提出了有益的意见：Anapum Bhide, Pete Chen, Ron David, Dave Ditzel, Fred Douglis, Dieter Gawlick, Jim Gray, Mark Hill, Doug Johnson, Joan Pendleton, Martin Schulze, 和 Herve Touati。这项工作得到了国家科学基金会的资助，根据MIP-5235准许。

**附件：可靠性计算**

利用概率理论，我们可以计算。我们首先假设独立和指数失效率。我们的模型使用了一个有偏差的硬币，硬币头部的概率是在第一次失败的平均修理时间（MTTR）内发生第二次失败的概率。由于磁盘故障是指数级的：

概率（至少有一个剩余磁盘在MTTR中出现故障）

在所有实际情况下

由于（）在0<X<<1时约为X：

概率（至少有一个剩余磁盘在MTTR中出现故障）

在磁盘故障时，我们掷硬币：

头=>系统崩溃，因为第二个故障发生在第一个故障被修复之前；

尾=>从错误中恢复并继续。

然后

在我们的模型中，群体故障并不是精确的指数，但是我们已经在实际的MTTR<<MTTF/（G+C）案例中验证了这个简化的假设。这使得整个系统的MTTF仅为除以组数 。

**参考文献**

[Bell 84] C.G. Bell, "The Mini and Micro Industries," *IEEE Computer,* Vol. 17, No. 10 (October 1984), pp. 14-30.

[Joy 85] B. Joy, presentation at ISSCC '85 panel session, Feb. 1985.

[Sicwiorck 82] D.P. Siewiorck, C.G. Bell, and A. Newell, *Computer Structures: Principles and Examples,* p. 46.

[Moore 75] G.E. Moore, "Progress in Digital Integrated Electronics,"*Proc. IEEE Digital Integrated Electronic Device Meeting,* (1975), p. 11.

[Myers 86] G.J. Myers, A.Y.C. Yu, and D.L. House, "Microprocessor Technology Trends," *Proc. IEEE,* Vol. 74, no. 12, (December 1986),pp. 1605-1622.

[Garcia 84] H. Garcia-Molina, R. Cullingford, P. Honeyman, R. Lipton,"The Case for Massive Memory," Technical Report 326, Dept. of EE and CS, Princeton Univ., May 1984.

[Myers 86] W. Myers, "The Competitiveness of the United States Disk Industry," *IEEE Computer,* Vol. 19, No. 11 (January 1986), pp. 85-90.

[Frank 87] P.D. Frank, "Advances in Head Technology," presentation at *Challenges in Disk Technology Short Course,* Institute for Information Storage Technology, Santa Clara University, Santa Clara, California, December 15-17, 1987.

[Stevens 81] L.D. Stevens, "The Evolution of Magnetic Storage," *IBM Journal of Research and Development,* Vol. 25, No.5, Sept. 1981, pp.663-675.

[Harker 81] J.M. Harker *et al.,* "A Quarter Century of Disk File Innovation," *ibid.,* pp. 677-689.

[Amdahl 67] G.M. Amdahl, "Validity of the single processor approach to achieving large scale computing capabilities," *Proceedings AFIPS 1967* *Spring Joint Computer Conference* Vol. 30 (Atlantic City, New Jersey April 1967), pp. 483-485.

[Bora 83] H. Boral and D.J. DeWitt, "Database Machines: An Ideas Whose Time Has Passed? A Critique of the Future of Database Machines," *Proc.1nternational Conf on Database Machines,* Edited by H.-0. Leilich and M. Misskoff, Springer-Verlag, Berlin, 1983.

[IBM 87] "IBM 3380 Direct Access Storage Introduction," IBM GC 26-4491-0, September 1987.

[Gawlick 87] D. Gawlick, private communication, Nov., 1987.

[Fujitsu 87] "M2361A Mini-Disk Drive Engineering Specifications," (revised) Feb., 1987, B03P-4825-0001A.

[Adaptec 87] AIC-6250, *IC Product Guide,* Adaptec, stock# DB0003-00 rev. B, 1987, p. 46.

[Livny 87] Livny, M., S. Khoshafian, H. Boral, "Multi-disk management algorithms," *Proc. of ACM SIGMETRICS,* May 1987.

[Kim 86] M.Y. Kim, "Synchronized disk interleaving," *IEEE Trans.on Computers,* vol. C-35, no. 11, Nov. 1986.

[Salem 86] K. Salem and Garcia-Molina, H., "Disk Striping," *IEEE 1986 Int. Conf on Data Engineering,* 1986.

[Bitton 88] D. Bitton and J. Gray, "Disk Shadowing," *in press,* 1988.

[Kurzweil88] F. Kurzweil, "Small Disk Arrays - The Emerging Approach to High Performance," presentation at Spring COMPCON 88, March 1, 1988, San Francisco, CA.

[Hamming 50] R. W. Hamming, "Error Detecting and Correcting Codes," *The Bell System Technical Journal,* Vol XXVI, No.2 (April 1950),pp. 147-160.

[Hillis 87] D. Hillis, private communication, October, 1987.

[Park 86] A. Park and K. Balasubramanian, "Providing Fault Tolerance in Parallel Secondary Storage Systems," Department of Computer Science, Princeton University, CS-TR-057-86, Nov. 7, 1986.

[Maginnis 87] N.B. Maginnis, "Store More, Spend Less: Mid-range Options *Abound,"Computerworld,* Nov. 16, 1987, p. 71.

[Denning 78] P.J. Denning and D.R. Slutz, "Generalized Working Sets for Segment Reference Strings," *CACM,* vol. 21, no. 9, (Sept. 1978) pp. 750-759.

[Bell 85] Bell, C.G., "Multis: a new class of multiprocessor computers,"Science, vol. 228 (April26, 1985) 462-467.