

使用高速缓存的另一种方法是在磁盘上保存每部电影的最初几分钟。这样，当一部电影被请求时，可以立刻从磁盘文件开始回放，与此同时，电影从DVD或磁带复制到磁盘上。通过始终在磁盘上存放电影足够长的部分，电影的下一个片断在它需要之前就已经取到磁盘上的概率会很高。如果一切都进行得很好，整部电影将在它需要之前就已经在磁盘上了，然后它将进入高速缓存并且停留在磁盘上以备随后有更多的请求。如果太多的时间过去而没有另外的请求，电影将从高速缓存中删除，以便为更为流行的电影腾出空间。

7.9 多媒体磁盘调度

多媒体对磁盘提出了与传统的、面向文本的应用程序（例如编译器或字处理器）有所不同的要求。特别是，多媒体要求极高的数据率和数据的实时传输。这些都不是轻易就能够提供的。此外，在视频服务器的情形中，让一个服务器同时处理几千个客户还存在着经济压力。这些需求影响着整个系统。上面我们了解了文件系统，现在让我们来看一看多媒体磁盘调度。

7.9.1 静态磁盘调度

尽管多媒体对系统的所有部分提出了巨大的实时和数据率要求，但是它还有一个特性使其比传统的系统更加容易处理，这就是可预测性。在传统的操作系统中，对磁盘块的请求是以相当不可预测的方式发出的。磁盘子系统所能做的最好不过是对每个打开的文件执行一个磁盘块的预读，除此之外，它能够做的全部事情就是等待请求的到来，并且在请求时对它们进行处理。多媒体就不同了，每个活动的视频流对系统施加明确的负载，使系统成为高度可预测的。就NTSC回放而言，每33.3ms，每个客户将需要其文件中的下一帧，并且系统有33.3ms的时间提供所有的帧（系统对每个视频流需要缓冲至少一帧，所以取第 $k+1$ 帧可以与第 k 帧的回放并行处理）。

这一可预测的负载可以用来使用为多媒体剪裁的算法对磁盘进行调度。下面我们将只考虑一个磁盘，但是其思想也可以运用于多个磁盘。就这个例子而言，我们将假设存在10个用户，每个用户观看不同的电影。此外，我们还将假设所有的电影具有相同的分辨率、帧率和其他特性。

根据系统的其他部分，计算机可能有10个进程，每个视频流一个进程，或者有一个具有10个线程的进程，或者甚至只有一个具有一个线程的进程，以轮转方式处理10个视频流。细节并不重要，重要的是，时间被分割成回环（round），在这里一个回环是一帧的时间（对于NTSC是33.3ms，对于PAL是40ms）。在每一回环的开始，为每个用户生成一个磁盘请求，如图7-26所示。

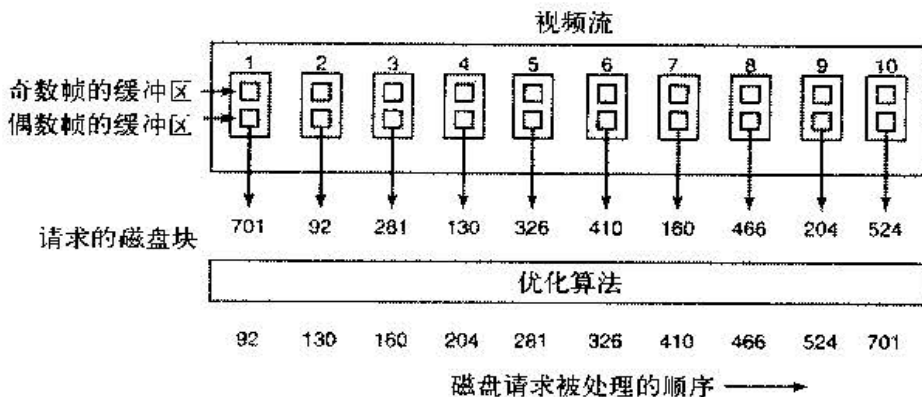


图7-26 在一个回环中，每部电影请求一帧

在回环的起始处，当所有的请求都进来之后，磁盘就会知道在那个回环期间它必须做什么，它还知道直到处理完这些请求并且下一个回环开始，不会有其他的请求进来。因此，它能够以优化的方法对请求排序，可能是以柱面顺序（可以想象在某些情形也可能以扇区顺序）排序，然后以优化的顺序对它们进行处理。在图7-26中，显示的请求是以柱面顺序排序的。

乍一看，人们可能会认为以这样的方式优化磁盘没有什么价值，因为只要磁盘满足最终时限，那么它是以1ms的富余满足还是以10ms的富余满足并没有什么关系。然而，这一结论是错误的。通过以这样的方式优化寻道，处理每一请求的平均时间就缩短了，这意味着一般来说每一回环磁盘可以处理更多的