高速缓存的文件片段称为视图(view),这是因为它们代表了被映射到文件系统的文件上的内核虚拟地址片段。所以,在高速缓存中,对物理内存的管理实际上是由内存管理器提供的。高速缓存管理器的作用是为视图管理内核虚拟地址的使用,命令内存管理器在物理内存中钉住页面,以及为文件系统提供接口。

Windows高速缓存管理器工具在文件系统中被广泛地共享。这是因为高速缓存是根据独立的文件来虚拟导址的,高速缓存管理器可以在文件的基础上很轻易地实现预读取。访问高速缓存数据的请求来自于每个文件系统。由于文件系统不需要先把文件的偏移转换成物理磁盘号然后再请求读取高速缓存的文件页,所以虚拟缓存非常方便。类似的转换发生在内存管理器调用文件系统访问存储在磁盘上的页面的时候。

除了对内核虚拟地址和用来缓存的物理内存资源的管理外,考虑到视图的一致性,大批量磁盘回写,以及文件结束标志的正确维护(特别是当文件扩展的时候),高速缓存管理器还必须与文件系统协作。在文件系统、高速缓存管理器和内存管理器之间管理文件最困难的方面在于文件中最后一个字节的偏移,即有效数据长度。如果一个程序写出了文件末尾,则越过的磁盘块都需要清零,同时为了安全的原因,在文件的元数据中记录的有效数据长度不应该允许访问未经初始化的磁盘块,所以全零磁盘块在文件元数据更新为新的长度之前必须写回到磁盘上。然而,可以预见的是,如果系统崩溃,一些文件的数据块可能还没有按照内存中的数据进行更新,还有一些数据块可能含有属于其他文件的数据,这都是不能接受的。

现在让我们来看看高速缓存管理器是如何工作的。当一个文件被引用时,高速缓存管理器映射一块大小为256KB的内核虚拟地址空间给文件。如果文件大于256KB,那么每次只有一部分文件被映射进来。如果高速缓存管理器耗尽了虚拟地址空间中大小为256KB的块,那么,它在映射一个新文件之前必须释放一个旧的文件。文件一旦被映射,高速缓存管理器通过把内核虚拟地址空间复制到用户缓冲区的方式来满足对该数据块的请求。如果要复制的数据块不在物理内存当中,会发生缺页中断,内存管理器会按照通常的方式处理该中断。高速缓存管理器甚至不知道一个数据块是不是在内存当中。复制总是成功的。

除了在内核和用户缓冲区之间复制的页面,高速缓存管理器也为映射到虚拟内存的页面和依靠指针访问的页面服务。当一个线程访问某一映射到文件中的虚拟地址但发生缺页的时候,内存管理器在大多数情况下能够使用软中断处理这种访问。如果该页面已经被高速缓存管理器映射到内存当中,即该页面已经在物理内存当中,那么就不需要去访问磁盘了。

高速缓存不一定适合所有的应用程序。大型企业应用程序,如SQL、希望自己來管理高速缓存和 I/O。Windows允许文件绕开高速缓存管理器,以未缓冲I/O的方式打开。从历史上看,这类应用程序使用一个可增长的用户态虚拟地址空间来替代操作系统提供的高速缓存,因此,系统应支持一种配置,使得重新启动后能给应用程序提供其所需的3GB的用户态地址空间,而只使用1GB的地址空间用于内核态來代替2-GB/2-GB的传统分割。这种运行模式(启动选项启用后,称为3GB模式)在一些允许以多种粒度来调整用户/内核地址空间分割的操作系统上不太灵活。当Windows运行在3GB模式下时,只有一半数量的内核虚拟地址可用。高速缓存管理器通过映射更少的文件来进行调整,这正是SQL所喜欢的。

Windows Vista在系统中引入了一种全新的、有别于高速缓存管理器的缓存技术,称为ReadyBoost。用户可以在USB接口或其他端口插入闪存,并命令操作系统使用闪存作为一个通写缓存。闪存引入了一种新的存储层次,这对于增加磁盘读缓存的数量特别有用。虽然比不上作为普通内存的动态RAM (DRAM),但是从闪存读取数据还是相当快的。结合高速的DRAM和相对廉价的闪存,Vista系统以少量的DRAM,使得不必开启机箱就可以获得更高的性能。

ReadyBoost压缩数据(通常为2倍),并加密。ReadyBoost使用一个过滤驱动程序来处理文件系统发送到卷管理器的I/O请求。名为ReadyBoot的类似技术,通过使用闪存缓存数据来加速Windows Vista系统的启动时间。但是这些技术对拥有1GB或更多内存的系统影响较小。在只有512MB内存的系统上尝试运行Windows Vista才是它们真正有帮助的地方。内存容量将近1GB的系统拥有足够的内存,页面请求非常罕见,使得磁盘I/O能够满足大多数使用场景。

通写方式对闪存被拔除时减少数据丢失很重要,但未来的PC硬件可能在主板上直接集成闪存。这样,闪存不用通写方式也可以使用,从而缓存系统故障时也需要继续存在的关键数据,而无需旋转磁盘。这不仅带来了性能的提升,而且还可以降低能耗(从而提高笔记本电脑的电池寿命),因为磁盘旋转少了。现在一些笔记本电脑一直在致力于使用大量的闪存来代替机电磁盘。