Revised notes on the reliability of FlushFileBuffers

devblogs.microsoft.com/oldnewthing/20170510-00

May 10, 2017

Raymond Chen

Some time ago, I wrote on the hard drives that lie about flushing file buffers (and the drivers who love them). Here's a check-in on what's happened over the past few years.

As things stand today, you can rely on `FlushFileBuffers` committing your changes to physical disk. (Noting of course that this operation is expensive and may not actually help you if you aren't updating your data transactionally.)

Historically, NTFS used the Force Unit Access flag to indicate to the driver that it should wait until the information is committed to non-volatile storage before signaling that the operation has completed. The conventional technique to force writes to go to media is to pass `FILE_FLAG_WRITE_THROUGH`, which internally tells the file system to set the FUA flag on I/O operations to that file. But as noted in this article, in the Windows 7 time frame, EIDE and SATA drivers do not respect the FUA flag, which means that your writes may still be buffered by the drive's internal memory.

Fortunately, even with those drivers, the `FlushFileBuffers` function will send the `FLUSH_CACHE` command, which tells the drive, "Hey, I know you have internal buffers. Look at me when I'm talking to you. Yes, you. With the internal buffers. Go flush them." Fortunately, nearly all drivers in the Windows 7 era respect that command. (There are a few stragglers that still ignore it.)

Fast-forward to Windows 8. Support for `FLUSH_CACHE` is required by all drives in order to be declared compatible with Windows 8, and the Windows storage team has worked with hard drive vendors to ensure that `FLUSH_CACHE` is properly respected. NTFS switched from using the FUA flag to the `FLUSH_CACHE` command to enforce consistency of its metadata.

Note that if an application opens a file with the `FILE_FLAG_WRITE_THROUGH` flag, NTFS still sets the FUA flag on the write operations. But as noted above, EIDE and SATA drivers do not respect the FUA flag, so asking for `FILE_FLAG_WRITE_THROUGH` doesn't give you any additional robustness in those cases.

If you want to flush the data to physical media, the `FlushFileBuffers` function is your ticket.

As noted in the linked article, there are options in the UI to disable the signal to flush intermediate buffers, but you should use them only if you have a separate UPS for the hard drive, or if you simply don't mind that the drive can get corrupted in the event of a power failure.

Raymond Chen

Follow