Flushing your performance down the drain, that is

devblogs.microsoft.com/oldnewthing/20100909-00

September 9, 2010

Raymond Chen

Some time ago, Larry Osterman discussed the severe performance consequences of flushing the registry, which is a specific case of the more general performance catch: Flushing anything will cost you dearly. A while back, I discussed the high cost of the "commit" function, and all the flush-type operations turn into a commit at the end of the day. ~~`FlushViewOfFile`~~ ; [see correction below] `FlushFileBuffers` , `RegFlushKey` , they all wait until the data has been confirmed written to the disk. If you perform one of these explicit flush operations, you aren't letting the disk cache do its job. These types of operations are necessary only if you're trying to maintain transactional integrity. If you're just flushing the data because "Well, I'm finished so I want to make sure it gets written out," then you're just wasting your (and the user's) time. The data will get written out, don't worry. Only if there is a power failure in the next two seconds will the data fail to get written out, but that's hardly a new problem for your program. If the power went out in the middle of the call to `FlushFileBuffers` (say, after it wrote out the data containing the new index but before it wrote out the data the index points to), you would've gotten partially-written data anyway. If you're not doing transactional work, then your call to `FlushFileBuffers` didn't actually fix anything. You still have a window during which inconsistency exists on the disk. Conclusion: View any call to ~~`FlushViewOfFile`~~ ; [see correction below] `FlushFileBuffers` , and `RegFlushKey` with great suspicion. They will kill your program's performance, and even in the cases in which you actually would want to call it, there are better ways of doing it nowadays. **More remarks on that old TechNet article**: The text for the *Enable advanced performance* check box has been changed in Windows 7 to something that more accurately describes what it does: *Turn off Windows write-cache buffer flushing on the device*. There's even explanatory text that explains the conditions under which it would be appropriate to enable that setting:

> To prevent data loss, do not select this check box unless the device has a separate power supply that allows the device to flush its buffer in case of power failure.

Hard drives nowadays are more than just platters of magnetic media. There's also RAM on the hard drive circuit board, and this RAM is used by the hard drive firmware as yet another buffer. If the drive is told, "Write this data to the hard drive at this location," the drive copies the data into its private RAM buffer and immediately returns a successful completion code to

the operating system. The drive then goes about seeking the head, looking for the sector, and physically writing out the data. When your program issues a write command to the file system (assuming that file system buffering is enabled), the write goes into the operating system disk cache, and periodically, the data from the operating system disk cache is flushed to the hard drive. As we saw above, the hard drive lies to the operating system and says "Yeah, I wrote it," even though it hasn't really done it yet. The data the operating system requested to be written is just sitting in a RAM buffer on the hard drive, that in turn gets flushed out to the physical medium by the hard drive firmware. If you call one of the `FlushBlahBlah` functions, Windows flushes out its disk cache buffers to the hard drive, as you would expect. But as we saw above, this only pushes the data into the RAM buffer on the hard drive. Windows understands this and follows up with another command to the hard drive, "Hey, I know you're one of those sneaky hard drives with an internal RAM buffer. Yes, I'm talking to you; don't act all innocent like. So do me a favor, and flush out your internal RAM buffers too, and let me know when that's done." This extra "I know what you did last summer" step ensures that the data really is on physical storage, and the `FlushBlahBlah` call waits until the "Okay, I finished flushing my internal RAM buffer" signal from the hard drive before returning control to your program. This extra "flush out your internal RAM buffer too" command is the right thing to do, but it can safely be skipped under *very special circumstances*: Consider a hard drive with a power supply separate from the computer which can keep the drive running long enough to flush out its internal RAM, even in the event of a sudden total loss of external power. For example, it might be an external drive with a separate power supply that is hooked up to a UPS. If you have this very special type of set-up, then Windows doesn't need to issue the "please flush out your internal RAM buffers too" command, because you have a guarantee that the data *will make it to the disk no matter what happens in the future*. Even if a transformer box explodes, cutting off all power to your building, that hard drive has enough residual power to get the data from the internal RAM buffer onto the physical medium. Only if your hard drive has that type of set-up is it safe to turn on the *Turn off Windows write-cache buffer flushing on the device* check box. (Note that a laptop computer battery does *not* count as a guarantee that the hard drive will have enough residual power to flush its RAM buffer to physical media. You might accidentally eject the battery out of your laptop, or you might let your battery run down completely. In these cases, the hard drive will not have a chance to finish flushing its internal RAM buffer.)

Of course, if the integrity of your disks is not important then go ahead and turn the setting on even though you don't have a battery backup. One case where this may be applicable is if you have a dedicated hard drive you don't care about losing if the power goes out. Many developers on the Windows team devote an entire hard drive to holding the files generated by a build of the operating system. Before starting a build, they reformat the drive. If the power goes out during a build, they'll just reformat the drive and kick off another build. In this case, go ahead and check the box that says *Enable advanced performance*. But if you care about the files on the drive, you shouldn't check the box unless you have that backup power supply.

Raymond Chen

Follow