If I issue multiple overlapped I/O requests against the same region of a file, will they execute in the order I issued them?

devblogs.microsoft.com/oldnewthing/20221111-00

November 11, 2022

Raymond Chen

Suppose you open a file in overlapped mode, allowing you to issue multiple overlapped I/O requests against the handle. You then issue two I/O write operations like this:

1. Write 8192 bytes starting at offset 0.
2. Write 8192 bytes starting at offset 4096.

What can be guaranteed about the contents of the file after both I/O operations have completed? Let's assume that 4096 is a multiple of the sector size, so all writes are to full sectors.

Can we assume that when everything is finished, the contents of the overlapping region come from the second write request?

No, that is not a valid assumption. The term *overlapped* in "overlapped I/O" refers to temporal overlap, and it means that multiple requests can be outstanding against a single handle, but each request has an indepedent lifetime. It is possible for one request to race ahead of another request that had been submitted earlier. This is a common occurrence for rotational storage media, because the driver will reorder I/O requests to reduce the amount of seeking required.

The driver is also likely to coalesce I/O operations that target the same physical sectors, and there is no requirement that overlapping writes be applied in the order of issue. It might be that the I/O passes through other layers before it reaches the disk driver, and the later write request might pass through the layer faster than the earlier one.

For example, in our brief look at the I/O layering, we see that I/O passes through the volume snapshot service, which can turn a single write I/O into a write, read, and write. In theory, then, this could happen:

First I/O (write to sector *N*)	**Second I/O (write to sector *N*)**
Check if sector is part of a lazy snapshot	
Yes: Need to back up original data	
Read sector *N*	
Reserve sector *N*′ for snapshot	
Write sector *N*′	
Mark sector *N* as not part of a lazy snapshot	
	Check if sector is part of a lazy snapshot
	No: Don’t need to back up original data
	Write sector *N*
Write sector *N*	

If a second write to sector *N* occurs at just the right time, it will race ahead of the first write, causing the two writes to reach the disk driver out of order.

Okay, so the writes can complete out of order to storage. But will they at least be atomic? Can I be sure that the final results will either be the entirety of the first write or the entirety of the second?

Not really.

If the write is spread out over multiple sectors, then the individual writes can get split. In the above example, if the write request involved two sectors, and only one of them was part of a lazy snapshot, then the other sector could complete first while the one that is part of the lazy snapshot deals with being backed up in the snapshot.

Okay, but if the write is a single sector, then will *those* be atomic? Am I sure that I will end up either with the entire sector from the first write or the entire sector from the second write?

I’m not so sure about even that.

So-called *Advanced Format* drives have a mode known as 512e where the firmware reports a sector size of 512 bytes, but the physical storage unit is something larger, like 4KB. In that case, writes that are not exact multiples of the physical storage unit are internally performed by the firmware as read-modify-write operations. You can run into pathological cases where a partition is created at an offset that is not an exact multiple of the physical storage unit size. Or you might have a VHD file (whose virtual sector size is always 512 bytes)[1] being stored on a host drive whose physical storage unit is 4KB.

I haven't worked out the math, but I wouldn't be surprised if there was some horrible pathological case where you have a misaligned VHD stored on a misaligned partition, mounted inside a virtual machine, hosted on a 4KB-native drive, and then exactly the wrong sequence of I/O operations is issued.

So just assume it can happen. If you need writes to complete in a specific order, then don't overlap them. Wait for the previous write to complete before issuing the next one.

Bonus chatter: If the target of the write isn't even a physical device but is instead something like an in-memory cache, then even more possibilities open up. In theory, you could even get byte-by-byte inconsistency, if the CPU scheduler is sufficiently devious.

[1] The VHDX virtual sector size is 4KB, so this sort of misalignment on Advanced Format drives is not going to happen for VHDX files, assuming the partition is 4KB-aligned.

Raymond Chen

Follow