

Windows 7 Support for Solid-State Drives

e7blog

There's a lot of excitement around the potential for the widespread adoption of solid-state drives (SSD) for primary storage, particularly on laptops and also among many folks in the server world. As with any new technology, as it is introduced we often need to revisit the assumptions baked into the overall system (OS, device support, applications) as a result of the performance characteristics of the technologies in use. This post looks at the way we have tuned Windows 7 to the current generation of SSDs. This is a rapidly moving area and we expect that there will continue to be ways we will tune Windows and we also expect the technology to continue to evolve, perhaps introducing new tradeoffs or challenging other underlying assumptions. Michael Fortin authored this post with help from many folks across the storage and fundamentals teams. --Steven

Many of today's Solid State Drives (SSDs) offer the promise of improved performance, more consistent responsiveness, increased battery life, superior ruggedness, quicker startup times, and noise and vibration reductions. With prices dropping precipitously, most analysts expect more and more PCs to be sold with SSDs in place of traditional rotating hard disk drives (HDDs).

In Windows 7, we've focused a number of our engineering efforts with SSD operating characteristics in mind. As a result, Windows 7's default behavior is to operate efficiently on SSDs without requiring any customer intervention. Before delving into how Windows 7's behavior is automatically tuned to work efficiently on SSDs, a brief overview of SSD operating characteristics is warranted.

Random Reads: A very good story for SSDs

SSDs tend to be very fast for random reads. Most SSDs thoroughly trounce traditionally HDDs because the mechanical work required to position a rotating disk head isn't required. As a result, the better SSDs can perform 4 KB random reads almost 100 times faster than the typical HDD (about 1/10th of a millisecond per read vs. roughly 10 milliseconds).

Sequential Reads and Writes: Also Good

Sequential read and write operations range between quite good to superb. Because flash chips can be configured in parallel and data spread across the chips, today's better SSDs can read sequentially at rates greater than 200 MB/s, which is close to double the rate many 7200 RPM drives can deliver. For sequential writes, we see some devices greatly exceeding the rates of typical HDDs, and most SSDs doing fairly well in comparison. In today's market, there are still considerable differences in sequential write rates between SSDs. Some greatly outperform the typical HDD, others lag by a bit, and a few are poor in comparison.

Random Writes & Flushes: Your mileage will vary greatly

The differences in sequential write rates are interesting to note, but for most users they won't make for as notable a difference in overall performance as random writes.

What's a long time for a random write? Well, an average HDD can typically move 4 KB random writes to its spinning media in 7 to 15 milliseconds, which has proven to be largely unacceptable. As a result, most HDDs come with 4, 8 or more megabytes of internal memory and attempt to cache small random writes rather than wait the full 7 to 15 milliseconds. When they do cache a write, they return success to the OS even though the bytes haven't been moved to the spinning media. We typically see these cached writes completing in a few hundred microseconds (so 10X, 20X or faster than actually writing to spinning media). In looking at millions of disk writes from thousands of telemetry traces, we observe 92% of 4 KB or smaller IOs taking less than 1 millisecond, 80% taking less than 600 microseconds, and an impressive 48% taking less than 200 microseconds. Caching works!

On occasion, we'll see HDDs struggle with bursts of random writes and flushes. Drives that cache too much for too long and then get caught with too much of a backlog of work to complete when a flush

Windows 7 Support for Solid-State Drives

e7blog

comes along, have proven to be problematic. These flushes and surrounding IOs can have considerably lengthened response times. We've seen some devices take a half second to a full second to complete individual IOs and take 10's of seconds to return to a more consistently responsive state. For the user, this can be awful to endure as responsiveness drops to painful levels. Think of it, the response time for a single I/O can range from 200 microseconds up to a whopping 1,000,000 microseconds (1 second).

When presented with realistic workloads, we see the worst of the SSDs producing very long IO times as well, as much as one half to one full second to complete individual random write and flush requests. This is abysmal for many workloads and can make the entire system feel choppy, unresponsive and sluggish.

Random Writes & Flushes: Why is this so hard?

For many, the notion that a purely electronic SSD can have more trouble with random writes than a traditional HDD seems hard to comprehend at first. After all, SSDs don't need to seek and position a disk head above a track on a rotating disk, so why would random writes present such a daunting a challenge?

The answer to this takes quite a bit of explaining, Anand's article admirably covers many of the details. We highly encourage motivated folks to take the time to read it as well as this fine USENIX paper. In an attempt to avoid covering too much of the same material, we'll just make a handful of points.

Most SSDs are comprised of flash cells (either SLC or MLC). It is possible to build SSDs out of DRAM. These can be extremely fast, but also very costly and power hungry. Since these are relatively rare, we'll focus our discussion on the much more popular NAND flash based SSDs. Future SSDs may take advantage of other nonvolatile memory technologies than flash.

A flash cell is really a trap, a trap for electrons and electrons don't like to be trapped. Consider this, if placing 100 electrons in a flash cell constitutes a bit value of 0, and fewer means the value is 1, then the controller logic may have to consider 80 to 120 as the acceptable range for a bit value of 0. A range is necessary because some electrons may escape the trap, others may fall into the trap when attempting to fill nearby cells, etc... As a result, some very sophisticated error correction logic is needed to insure data integrity.

Flash chips tend to be organized in complex arrangements, such as blocks, dies, planes and packages. The size, arrangement, parallelism, wear, interconnects and transfer speed characteristics of which can and do vary greatly.

Flash cells need to be erased before they can be written. You simply can't trust that a flash cell has no residual electrons in it before use, so cells need to be erased before filling with electrons. Erasing is done on a large scale. You don't erase a cell; rather you erase a large block of cells (like 128 KB worth). Erase times are typically long -- a millisecond or more.

Flash wears out. At some point, a flash cell simply stops working as a trap for electrons. If frequently updated data (e.g., a file system log file) was always stored in the same cells, those cells would wear out more quickly than cells containing read-mostly data. Wear leveling logic is employed by flash controller firmware to spread out writes across a device's full set of cells. If done properly, most devices will last years under normal desktop/laptop workloads.

It takes some pretty clever device physicists and some solid engineering to trap electrons at high speed, to do so without errors, and to keep the devices from wearing out unevenly. Not all SSD manufacturers are as far along as others in figuring out how to do this well.

Windows 7 Support for Solid-State Drives

e7blog

Performance Degradation Over Time, Wear, and Trim

As mentioned above, flash blocks and cells need to be erased before new bytes can be written to them. As a result, newly purchased devices (with all flash blocks pre-erased) can perform notably better at purchase time than after considerable use. While we've observed this performance degradation ourselves, we do not consider this to be a show stopper. In fact, except via benchmarking measurements, we don't expect users to notice the drop during normal use.

Of course, device manufactures and Microsoft want to maintain superior performance characteristics as best we can. One can easily imagine the better SSD manufacturers attempting to overcome the aging issues by pre-erasing blocks so the performance penalty is largely unrealized during normal use, or by maintaining a large enough spare area to store short bursts of writes. SSD drives designed for the enterprise may have as high as 50% of their space reserved in order to provide lengthy periods of high sustained write performance.

In addition to the above, Microsoft and SSD manufacturers are adopting the Trim operation. In Windows 7, if an SSD reports it supports the Trim attribute of the ATA protocol's Data Set Management command, the NTFS file system will request the ATA driver to issue the new operation to the device when files are deleted and it is safe to erase the SSD pages backing the files. With this information, an SSD can plan to erase the relevant blocks opportunistically (and lazily) in the hope that subsequent writes will not require a blocking erase operation since erased pages are available for reuse.

As an added benefit, the Trim operation can help SSDs reduce wear by eliminating the need for many merge operations to occur. As an example, consider a single 128 KB SSD block that contained a 128 KB file. If the file is deleted and a Trim operation is requested, then the SSD can avoid having to mix bytes from the SSD block with any other bytes that are subsequently written to that block. This reduces wear.

Windows 7 requests the Trim operation for more than just file delete operations. The Trim operation is fully integrated with partition- and volume-level commands like Format and Delete, with file system commands relating to truncate and compression, and with the System Restore (aka Volume Snapshot) feature.

Windows 7 Optimizations and Default Behavior Summary

As noted above, all of today's SSDs have considerable work to do when presented with disk writes and disk flushes. Windows 7 tends to perform well on today's SSDs, in part, because we made many engineering changes to reduce the frequency of writes and flushes. This benefits traditional HDDs as well, but is particularly helpful on today's SSDs.

Windows 7 will disable disk defragmentation on SSD system drives. Because SSDs perform extremely well on random read operations, defragmenting files isn't helpful enough to warrant the added disk writing defragmentation produces. The FAQ section below has some additional details.

By default, Windows 7 will disable Superfetch, ReadyBoost, as well as boot and application launch prefetching on SSDs with good random read, random write and flush performance. These technologies were all designed to improve performance on traditional HDDs, where random read performance could easily be a major bottleneck. See the FAQ section for more details.

Since SSDs tend to perform at their best when the operating system's partitions are created with the SSD's alignment needs in mind, all of the partition-creating tools in Windows 7 place newly created partitions with the appropriate alignment.

Windows 7 Support for Solid-State Drives

e7blog

Frequently Asked Questions

Before addressing some frequently asked questions, we'd like to remind everyone that we believe the future of SSDs in mobile and desktop PCs (as well as enterprise servers) looks very bright to us. SSDs can deliver on the promise of improved performance, more consistent responsiveness, increased battery life, superior ruggedness, quicker startup times, and noise and vibration reductions. With prices steadily dropping and quality on the rise, we expect more and more PCs to be sold with SSDs in place of traditional rotating HDDs. With that in mind, we focused an appropriate amount of our engineering efforts towards insuring Windows 7 users have great experiences on SSDs.

Will Windows 7 support Trim?

Yes. See the above section for details.

Will disk defragmentation be disabled by default on SSDs?

Yes. The automatic scheduling of defragmentation will exclude partitions on devices that declare themselves as SSDs. Additionally, if the system disk has random read performance characteristics above the threshold of 8 MB/sec, then it too will be excluded. The threshold was determined by internal analysis.

The random read threshold test was added to the final product to address the fact that few SSDs on the market today properly identify themselves as SSDs. 8 MB/sec is a relatively conservative rate. While none of our tested HDDs could approach 8 MB/sec, all of our tested SSDs exceeded that threshold. SSD performance ranged between 11 MB/sec and 130 MB/sec. Of the 182 HDDs tested, only 6 configurations managed to exceed 2 MB/sec on our random read test. The other 176 ranged between 0.8 MB/sec and 1.6 MB/sec.

Will Superfetch be disabled on SSDs?

Yes, for most systems with SSDs.

If the system disk is an SSD, and the SSD performs adequately on random reads and doesn't have glaring performance issues with random writes or flushes, then Superfetch, boot prefetching, application launch prefetching, ReadyBoost and ReadDrive will all be disabled.

Initially, we had configured all of these features to be off on all SSDs, but we encountered sizable performance regressions on some systems. In root causing those regressions, we found that some first generation SSDs had severe enough random write and flush problems that ultimately lead to disk reads being blocked for long periods of time. With Superfetch and other prefetching re-enabled, performance on key scenarios was markedly improved.

Is NTFS Compression of Files and Directories recommended on SSDs?

Compressing files help save space, but the effort of compressing and decompressing requires extra CPU cycles and therefore power on mobile systems. That said, for infrequently modified directories and files, compression is a fine way to conserve valuable SSD space and can be a good tradeoff if space is truly a premium.

We do not, however, recommend compressing files or directories that will be written to with great frequency. Your Documents directory and files are likely to be fine, but temporary internet directories or mail folder directories aren't such a good idea because they get large number of file writes in bursts.

Does the Windows Search Indexer operate differently on SSDs?

No.

Windows 7 Support for Solid-State Drives

e7blog

Is Bitlocker's encryption process optimized to work on SSDs?

Yes, on NTFS. When Bitlocker is first configured on a partition, the entire partition is read, encrypted and written back out. As this is done, the NTFS file system will issue Trim commands to help the SSD optimize its behavior.

We do encourage users concerned about their data privacy and protection to enable Bitlocker on their drives, including SSDs.

Does Media Center do anything special when configured on SSDs?

No. While SSDs do have advantages over traditional HDDs, SSDs are more costly per GB than their HDD counterparts. For most users, a HDD optimized for media recording is a better choice, as media recording and playback workloads are largely sequential in nature.

Does Write Caching make sense on SSDs and does Windows 7 do anything special if an SSD supports write caching?

Some SSD manufacturers including RAM in their devices for more than just their control logic; they are mimicking the behavior of traditional disks by caching writes, and possibly reads. For devices that do cache writes in volatile memory, Windows 7 expects flush commands and write-ordering to be preserved to at least the same degree as traditional rotating disks. Additionally, Windows 7 expects user settings that disable write caching to be honored by write caching SSDs just as they are on traditional disks.

Do RAID configurations make sense with SSDs?

Yes. The reliability and performance benefits one can obtain via HDD RAID configurations can be had with SSD RAID configurations.

Should the pagefile be placed on SSDs?

Yes. Most pagefile operations are small random reads or larger sequential writes, both of which are types of operations that SSDs handle well.

In looking at telemetry data from thousands of traces and focusing on pagefile reads and writes, we find that

Pagefile.sys reads outnumber pagefile.sys writes by about 40 to 1,

Pagefile.sys read sizes are typically quite small, with 67% less than or equal to 4 KB, and 88% less than 16 KB.

Pagefile.sys writes are relatively large, with 62% greater than or equal to 128 KB and 45% being exactly 1 MB in size.

In fact, given typical pagefile reference patterns and the favorable performance characteristics SSDs have on those patterns, there are few files better than the pagefile to place on an SSD.

Are there any concerns regarding the Hibernation file and SSDs?

No, hiberfile.sys is written to and read from sequentially and in large chunks, and thus can be placed on either HDDs or SSDs.

What Windows Experience Index changes were made to address SSD performance characteristics?

In Windows 7, there are new random read, random write and flush assessments. Better SSDs can score above 6.5 all the way to 7.9. To be included in that range, an SSD has to have outstanding random read rates and be resilient to flush and random write workloads.

Windows 7 Support for Solid-State Drives

e7blog

In the Beta timeframe of Windows 7, there was a capping of scores at 1.9, 2.9 or the like if a disk (SSD or HDD) didn't perform adequately when confronted with our random write and flush assessments. Feedback on this was pretty consistent, with most feeling the level of capping to be excessive. As a result, we now simply restrict SSDs with performance issues from joining the newly added 6.0+ and 7.0+ ranges. SSDs that are not solid performers across all assessments effectively get scored in a manner similar to what they would have been in Windows Vista, gaining no Win7 boost for great random read performance.