Touch Rate

A metric for analyzing storage system performance

By Steven Hetzler and Tom Coughlin

Introduction

Motivation

We have created a scale independent method for classifying the performance of storage systems under differing workloads. The goal was to find something that could be used as a both a rule of thumb as well as for detailed analysis. We wanted this analysis to be easy to understand and be useful for comparing different system designs. Further, it should reflect the behavior of the system as it is being used, not when it is idle (unless that is how it will be used). We call this new metric "Touch Rate".

Response time definition

Since we want Touch Rate to be a measure for a busy system operating at load, we need to define what is meant to be "at load." We define the load as back-to-back input/output operations (IOs), which is 100% utilization without queuing. We leave queuing aside for this study as it makes the analysis simpler. (See the appendix for details on touch rate terminology.)

The response time is the time to complete an IO operation, including the transfer of data and restoring the system for a subsequent IO operation. The response time is therefore a function of the IO object size as well as the speed of ancillary support operations. This is distinct from the access time, which is the time to the first byte of data after a request for an idle system.

The response time is thus a measure of how rapidly an object can be retrieved under operating conditions.

Touch rate definition

Touch rate is defined as the portion of the total capacity of a system that can be accessed in a given interval of time. In financial terms, it can be thought of as the number of inventory turns on the data set. This analogy leads us to look at touch rate as a measure of the value that can be extracted from the data set. We need to pick a time interval for measuring the touch rate that is appropriate to an application – for instance, a year is a suitable period for archival data.

Equation 1 gives a definition of the touch rate over a year.

$$\boldsymbol{Touch/Y} = \frac{\boldsymbol{ObjectSize}(\boldsymbol{MB}) \times \mathbf{1000}}{\boldsymbol{ResponseTime}(\boldsymbol{s}) \times \boldsymbol{Capacity}(\boldsymbol{TB}) \times \mathbf{31.5}} \qquad \textbf{Equation 1}$$

The above equation assumes the object size is in MB and the system or unit capacity is in TB. *Response_time* is the steady state response time in seconds (as described above) for this object size undergoing back-to-back 100% random IOs.

It is commonly stated that most systems will access only a fraction of the bulk data stored in a year, such as 10%. This implies that the required touch rate is 0.1/Y. More details on computing the touch rate are given in the appendix.

We might pick a shorter interval than 1 year for high performance data, and such a conversion is straightforward. Some examples are shown below,

$$Touch/Day = \frac{Touch/Y}{365},$$

$$Touch/Hour = \frac{Touch/Y}{8760}.$$

Note that a touch rate greater than one doesn't necessarily mean that the same data is accessed repeatedly, although it may. It can also mean that new data is coming in, which is also counted in the touch rate. What matters here is the amount of data accessed.

The touch rate is thus a measure of how much of the data in a system can be accessed during a given interval of time.

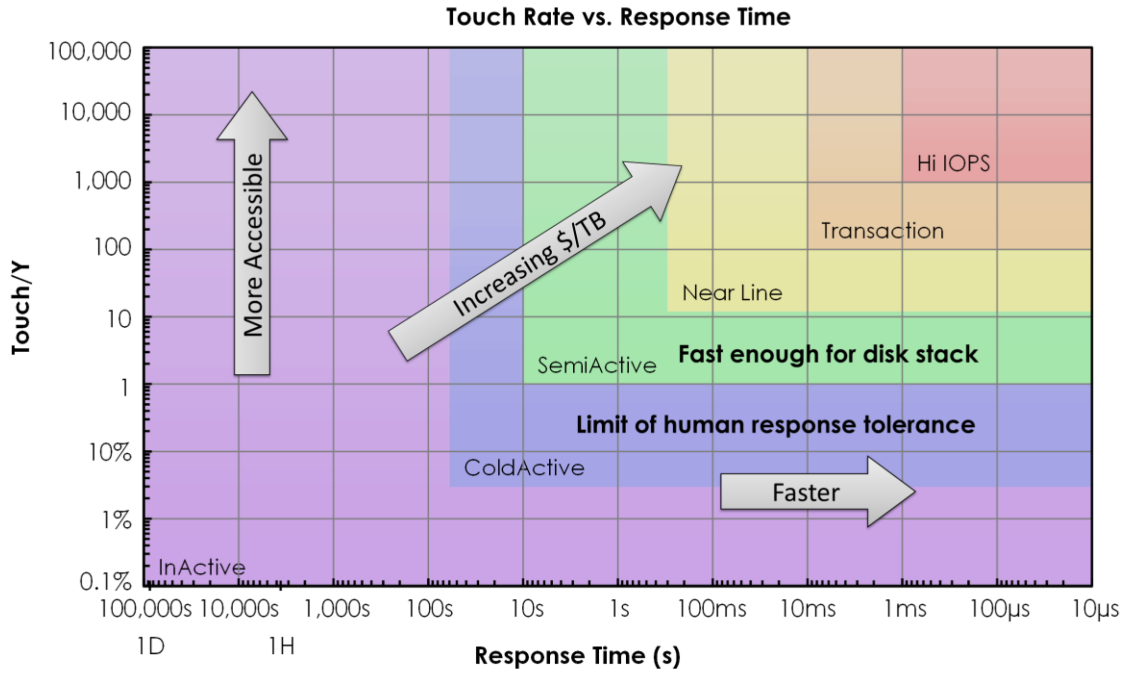
The Touch Rate Chart

Touch rate vs. response time

We can learn a lot about the behavior of a system by plotting the touch rate vs. the response time. The response time measures the time to act on a single object, while the touch rate relates to the time to act upon the data set as a whole. **Figure 1** shows such a chart that includes some indications of various sorts of applications (performance regions) and general trade-offs important to system design indicated. Note that this is a log-log chart with log base-10 scales on both the vertical and horizontal axes.

The chart shows touch rate as log touch per year on the vertical axis, and log response time on the horizontal axis with faster response times on the left. Shorter response time means data can be accessed more quickly, increasing its value. Higher touch rate means more data can be processed in a given time period, increasing the value that can be extracted. Thus, the data value increases for objects to the upper right. However, system costs also tend to increase to the upper right. Note that the total value of the data includes the amount of data, so most of the data value could be at lower performance regions.

Figure 1. Touch rate versus response time indicating various types of uses



We have defined six performance regions as shown in **Table 1** and **Figure 1.** While the boundaries are somewhat arbitrary, they are representative of the performance bounds for many workloads.

Table 1. Six performance regions for a discussion of storage system design

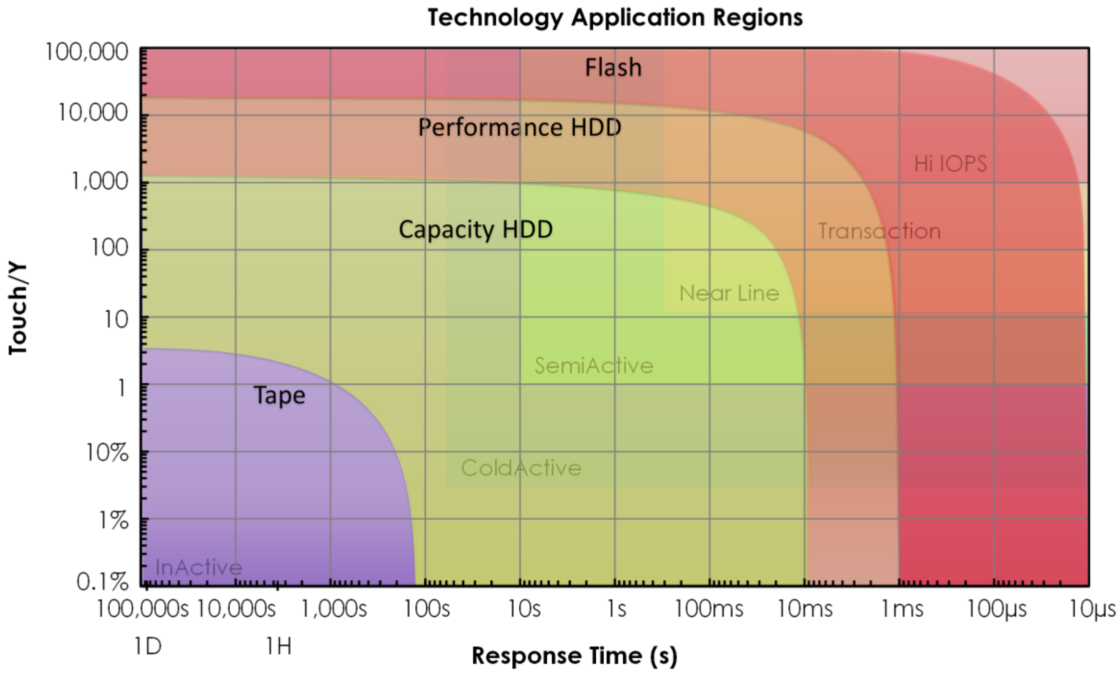
Region	Minimum response time	Minimum touch/Y
High (Hi) IOPS	1 ms	1,000 (~3/day)
Transaction	10 ms	100 (~1 per 3 days)
Near Line	300 ms	12 (1/month)
Semi-active	10 s (HDD SW stack)	1
Cold-active	60 s (human tolerance)	3%
InActive	> 60s	< 3%

The Hi IOPS region is for storage system performance beyond the capabilities of HDDs and may require flash memory or even volatile RAM. The touch rate lower bound for the Hi IOPS region is near 3/day. The Transaction region is for transaction data, and the performance limits are those of high performance enterprise disks. The touch rate lower bound for the transaction region is about 1 every 3 days (or 0.33/day). The Near Line region has performance limits matching the characteristics of high capacity hard disk drives. The touch rate limit is 1 per month. The semi-active region is defined as having a response time fast enough to not have time-outs when used with an HDD software stack (typically about 10 seconds). A system in this region would then be functionally plug-compatible with an HDD-based system, albeit slower. The touch rate lower limit for the semi-active region is 1 per year. The Cold-Active region is characteristic of on-line archives with human interaction. Thus, the response time limit is 60 seconds, which is about the human tolerance for an IO operation. If the IOs take longer than this, most humans would assume the IO has failed or wander off and do something else. The touch rate lower bound here is 3% per year. Everything slower and below 3% touch/Y is in the InActive archive region (the slowest response times in Figure 1).

Technology regions

Figure 2 shows where various storage technologies lie in the touch rate chart as shown in Figure 1. Flash, HDD and tape are used to address workloads in different regions. These storage technology regions have a hockey stick shape due to the way performance limits impact the system performance. On the right side of a region, the performance for small objects is dominated by the access time (time to get to the first byte of data). On the left side, the performance for larger objects is dominated by the data transfer time.

Figure 2. Digital storage technologies regions overlaid on the Touch Rate/Response Time chart



A given technology is most cost effective at the upper right boundary (the knee of the storage technology curve), where its full performance can be extracted. It becomes less cost effective as it used more to the lower left of this knee. At some point as the requirements move to the left, a lower performing technology is usually more cost effective.

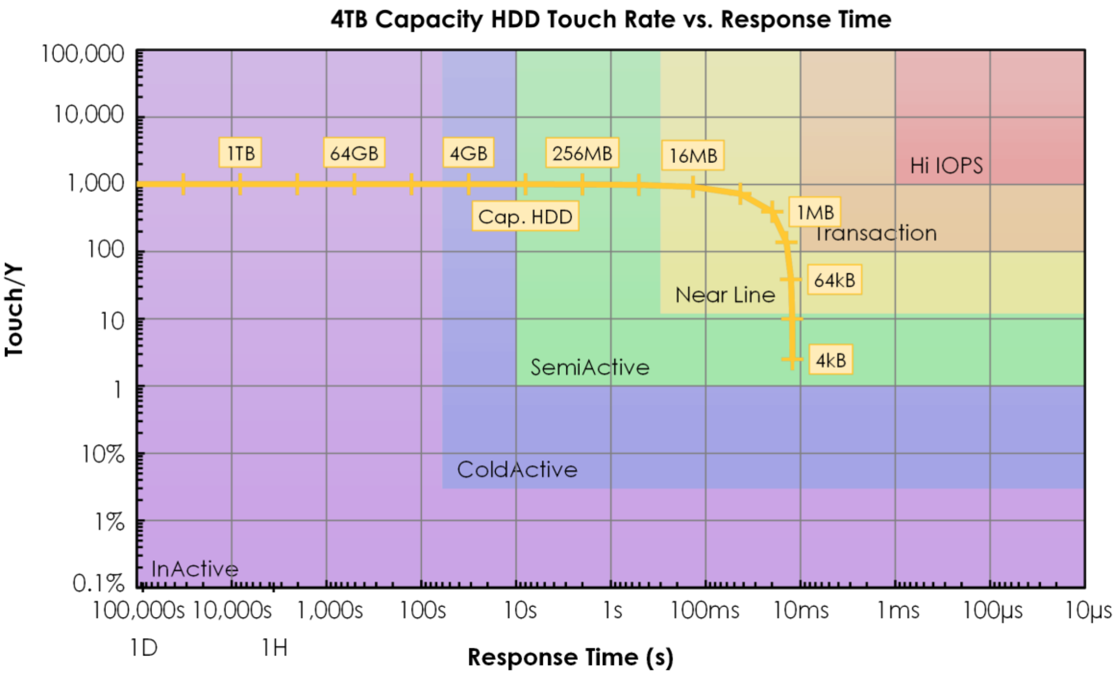
Moving the performance of a storage technology to the upper right, beyond its native capability is difficult, and is usually a very expensive proposition. It frequently involves reducing the capacity of each storage unit by overprovisioning. For example, short stroking HDDs can reduce the response time and increase the touch rate, but it also significantly increases the $/TB. If the capacity is limited to 25% of the total available HDD capacity, the touch rate will go up by 4x, but so will the $/TB. In a removable media library, increasing the ratio of drives to storage media similarly increases the touch rate.

The touch rate chart can be extended up beyond flash, and we anticipate that new non-volatile storage technologies will enter this region in the not too distant future.

IO Object size curve

The object size used by an application influences both the touch rate and the response time. Thus, we will get a curve for the touch rate vs. response time as a function of the object size for various storage technologies. The object size is indicated along this curve. The curve shown in **Figure 3** is for 100% random IO at the given object sizes for a typical 4TB capacity HDD system.

Figure 3. Touch/Y and response time for 100% random IO in a 4 TB capacity HDD



The touch rate versus response time curve for this capacity HDD has a shape similar to that shown for technologies in the technology regions chart (Figure 2). At a 4kB object size, the response time is 12ms and touch rate is 2.5/Y. While this is a near line drive, this small object size falls outside the near line performance region. 4kB is an object size associated with transaction workloads, and the curve shows why capacity HDDs aren't favored for such workloads.

At a 64kB object size, however, the touch rate has grown to 39/Y, and the response time has only increased slightly to 12.6ms. Thus, this is a better object size for this class of HDD, and is a more typical object size for near line workloads. At a 1 MB object size, the touch rate has reached 400/Y, and the response time has increased to 20ms. By this object size we are reaching the point where the data transfer time component of the response time is becoming important.

At a 64MB object size the touch rate has nearly saturated at 990/Y, but the response time has now grown to 530ms. Beyond this, the touch rate is essentially constant, and the response time grows linearly with the object size. We use the term *saturation touch rate* to refer to this region with constant touch rate over a wide range of response times.

When reading the touch rate chart, the object size chosen should be suitable for the application being considered. Similarly, when comparing different systems for a given application, the touch rate and response time values should be compared at the same object size.

So long as the system has enough bandwidth to operate all the devices simultaneously, the touch rate curve is independent of the number of storage devices being used. Thus, the touch rate, in this case, is scale independent. As a consequence, the touch rate curve is the same for one device as it is for a system of such devices.

If the system has resource limits (such as available network bandwidth), then full scaling may not be achieved, and the touch rate could be lower. If the data is striped in parallel across the devices, the result is a shifting of the object size points along the curve. Striping increases the data rate (and thus decreases the response time), but decreases the effective block size on a device (it spreads the block over multiple devices). Thus, it can be thought of as moving the object size locations on the curve toward the right and down.

Caching can also alter the system performance. The device charts also don't include system caching effects. We will discuss how to analyze caching later.

There may be separate curves for read and write operations. For example, flash memory devices tend to have different performance characteristics for reading and writing information.

The touch rate chart is useful for determining other system behaviors. For example, the response time for a 1TB object here is 1.7 hours. This means that a full drive scan on this 4TB drive will take at least 1.7x4 = 7 hours to read or write the full capacity of the drive. There are many tasks which do this: replication. RAID rebuild, full data scrub, etc.

High Performance Storage

Figure 4 shows the touch rate for a system comprising high performance 600GB 15krpm HDDs. The curve is significantly above and to the right of the capacity HDD curve, which is expected. At a 4kB object size, the touch rate is 42/Y and the response time 5ms. Like the capacity HDD, the touch rate begins to increase around a 1MB object size, and saturates around a 64MB object size.

The 4kB object touch rate is a bit below the transaction-processing region, which commonly uses this object size. As mentioned above, we have not included the effects from command reordering with queuing, which would improve the touch rate at the expense of the response time. Further, a 1/3 short stroke (using only the outer 1/3 of the disk to hold data) would give a touch rate of 126/Y, which is well within the transaction region. This technique is quite common, but comes at a steep $/TB price. This may be part of the reason flash has made significant inroads into this space.

The saturation touch rate for this high performance HDD is around 11,000/Y.

Figure 4. Touch/Y and response time for 100% random IO for a 600 GB 15k RPM HDD

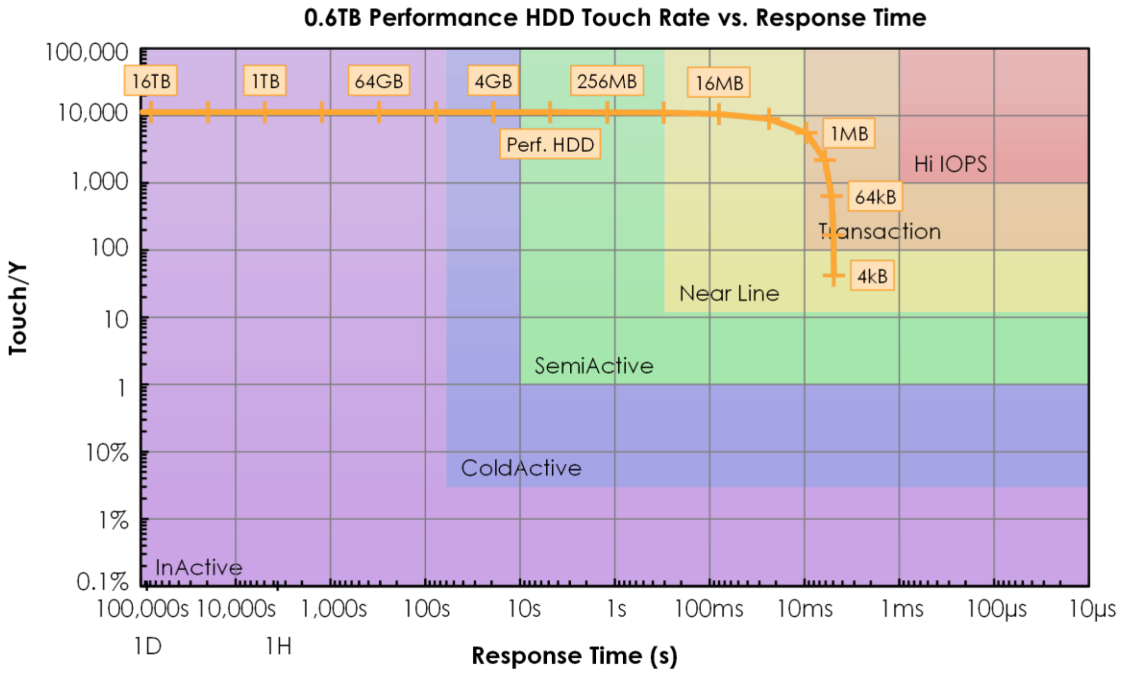


Figure 5 shows the touch rate for enterprise–grade SSDs when reading. The value of SSDs is readily apparent in the response time shift for small block IO. The 4kB response time is 65 times faster than the performance HDD, and has a 33 times greater touch rate. The saturation touch rate of 27,000/Y is only about 2.4 times better than the performance HDD. Thus, SSD is used primarily for small block size workloads.

Figure 6 shows the touch rate for enterprise–grade PCIe attached flash, which is faster than SATA or SAS SSDs. This curve is for read operations. PCIe flash outperforms traditional SSDs across the board, with about 2x faster response time and 2x greater touch rate at common block sizes.

Figure 5. Touch/Y and response time for reading for an enterprise SSD during read

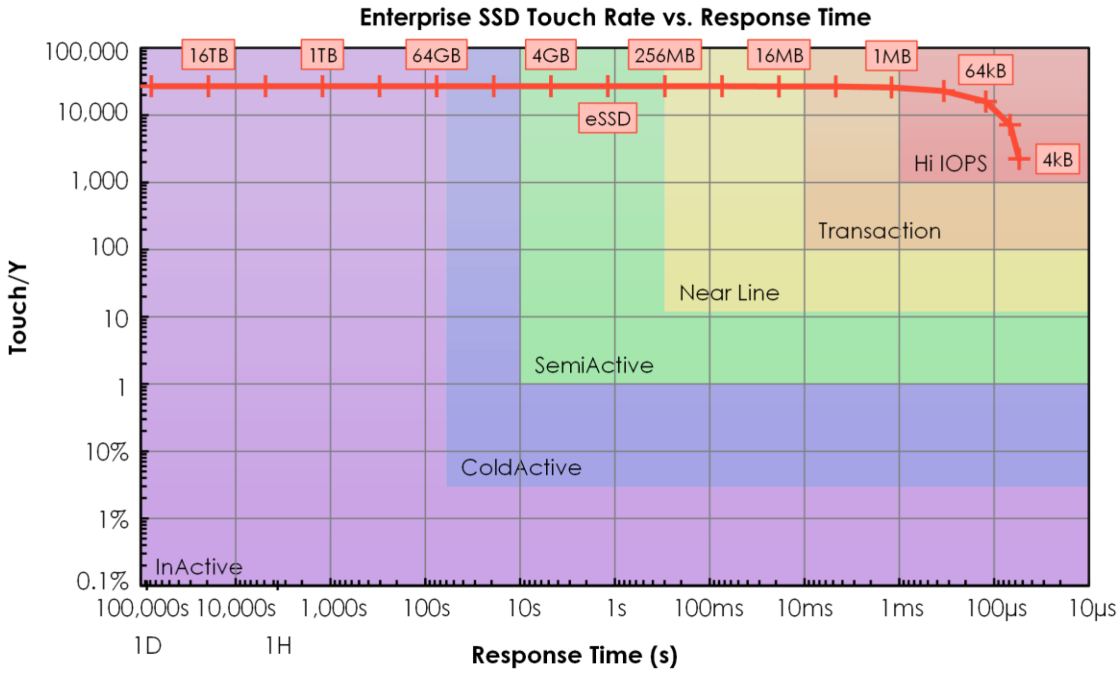
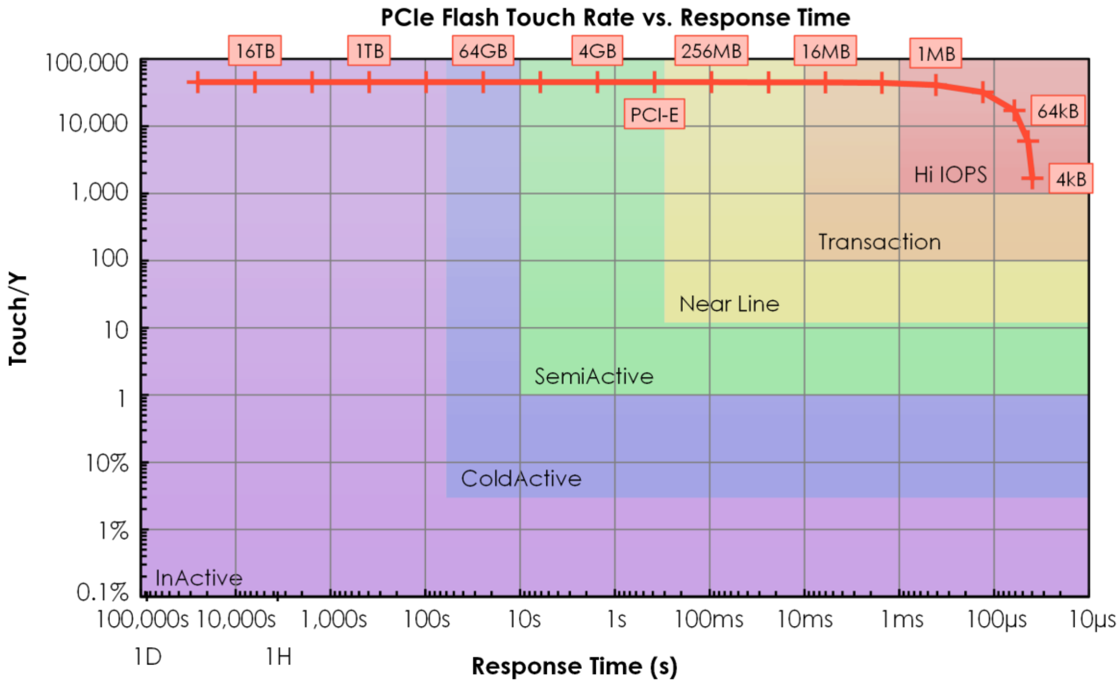


Figure 6. Touch/Y and response time for reading for an enterprise PCIe device



The touch rate charts highlight and help quantify the achievable benefits from various flash storage devices.

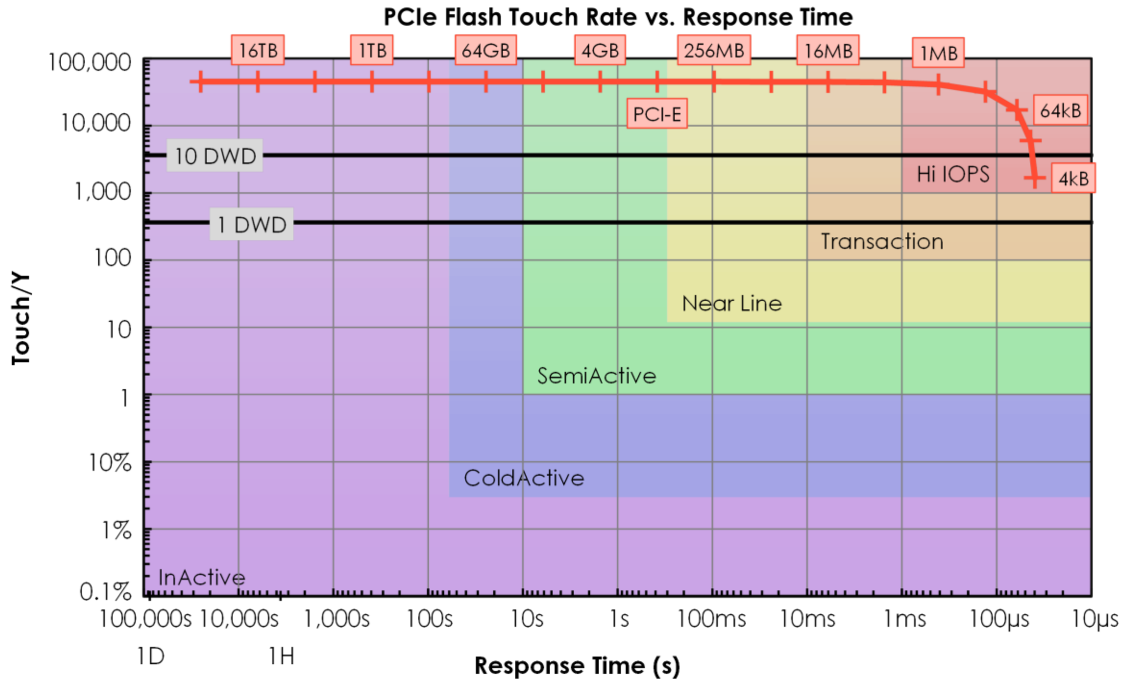
Limits on Touch rate

The touch rate charts shown for flash storage were for read operations only. Write performance generally differs from read performance for flash storage (it is often slower). The slower writes are the result of the erase/write requirement for writing flash memory and may be exaggerated by flash garbage collection and flash wear leveling.

One key restriction for flash write performance is the result of the limited write endurance of the flash devices. This restriction is commonly expressed for flash device as a drive writes per day (DWD) limit. Drive writes per day is itself a write touch rate (Touch/Y = 365* DWD). Enterprise SSDs commonly support 10-25 DWD[1] and PCI-e flash devices are often lower than 5 DWD[2].

Figure 7 shows touch rate limit lines for DWD specifications of 1 and 10. Thus, sustained touch rates above these lines will result in early failure of the devices. The limits most severely impact the touch rate at large object sizes, which are not the predominant workloads for flash devices. The touch rate chart makes the impact clear.

Figure 7. Flash DWD limits expressed in a touch rate chart



[1] For instance the HGST Ultrastar SSD1600MM and Ultrastar SSD800MH.B have DWD of 10 and 25 respectively, http://www.hgst.com/solid-state-storage/enterprise-ssd

[2] For example the Fusion ioDrive2 has a DWD of 4.4

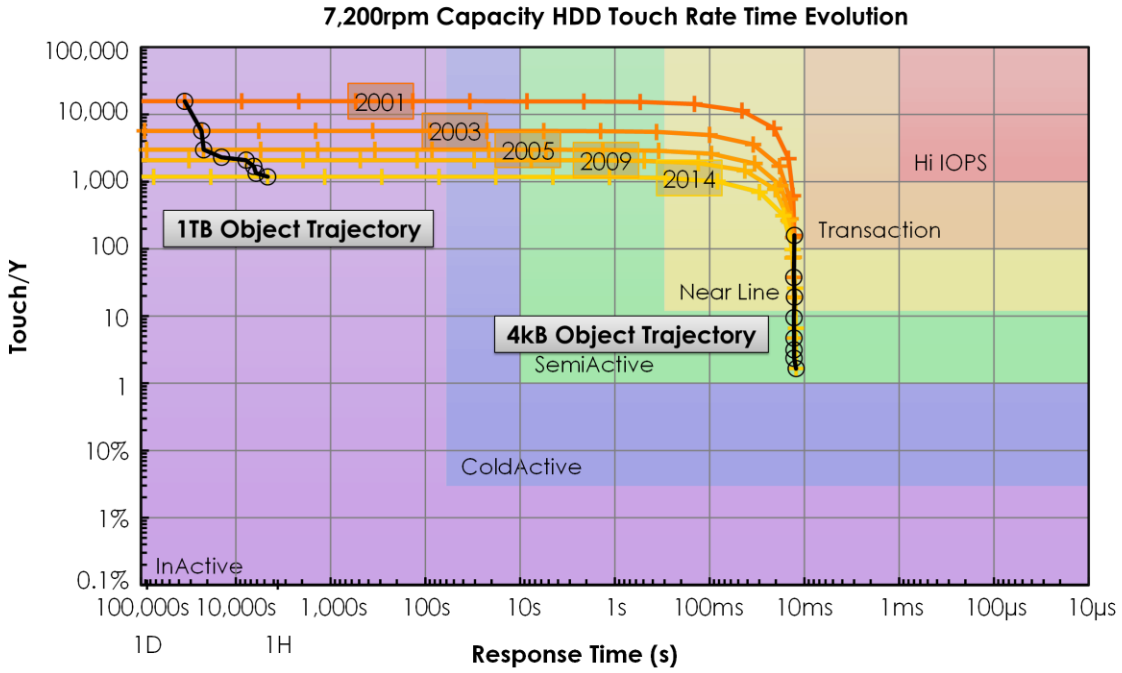
Flash devices aren't alone in having touch rate limits. Many HDDs today are specified with data transfer limits[3]. Typically, these are specified per year, which again can be easily converted to touch rate limits. In the case of HDDs, there is no known wearout mechanism associated with transferring data. So it is likely that limits exist to protect gross-margin wearout, as they provide a means to distinguish between otherwise similar drives. These are then reflected in the warranty.

Time Evolution

In general, touch rates decline as storage capacities increase for a given device or media technology. In order for the touch rate to remain constant, the data transfer rate would have to scale proportionally with capacity. Unfortunately, this tends not to be the case for most storage technologies.

Figure 8 shows Touch Rate curves for 7,200rpm capacity HDDs from 2001 through 2014.

Figure 8. Touch/Y for enterprise HDD from 2001 to 2014 and projections to 2024



The touch rate curves shift down each generation, which we like to call *the tyranny of density*. In HDDs, the touch rate is reduced due to increases in capacity with areal density increases, since they generally increase their capacity faster than their data rate. This is highlighted in the 1TB object trajectory. Some of the areal density

[3] Seagate Barracuda SATA Product Manual, Gen 14, Rev. G, October 2012

increase is due to linear density improvement, which increases the data transfer rate. When this occurs, the object point on a curve moves to the right and down along the touch rate curve. This is clear in the 1TB object trajectory. The capacity is also increased by track density improvements, which don't impact the data transfer rate. Thus, the touch rate declines.

In 2001, the capacity was 60GB, the 1TB object had a response time of 35,000s and a touch rate of 16,000/Y. By 2014, the capacity was 6TB (100x!) the 1TB object response time had dropped to 2,600s (7.5x) and the touch rate had dropped to 1,200/Y (13.3x). Note that 7.5 x 13.3 = 100, which it should.

The 4kB object trajectory is essentially vertical here, showing that there were limited improvements in the access time. In 2001, the 4kB touch rate was 160/Y. By 2014 it had dropped to 1.6/Y – the entire 100x.

Lately, we have seen a reduction in the areal density growth rates. This has led to the resurgence of platter stacking to increase HDD capacity. However, this causes a proportional reduction in the touch rate, as it doesn't improve the data transfer rate. (In other words, the curve just shifts straight downward.)

Scaling mismatches create opportunities for new technologies to fill in the vacated higher touch rate space. This is one of the factors contributing to the growth of flash in the high IOPS region and transaction regions. Flash has been able to displace enterprise HDD because the value extracted from the faster response time and higher touch rate exceeded the greater cost of flash memory.

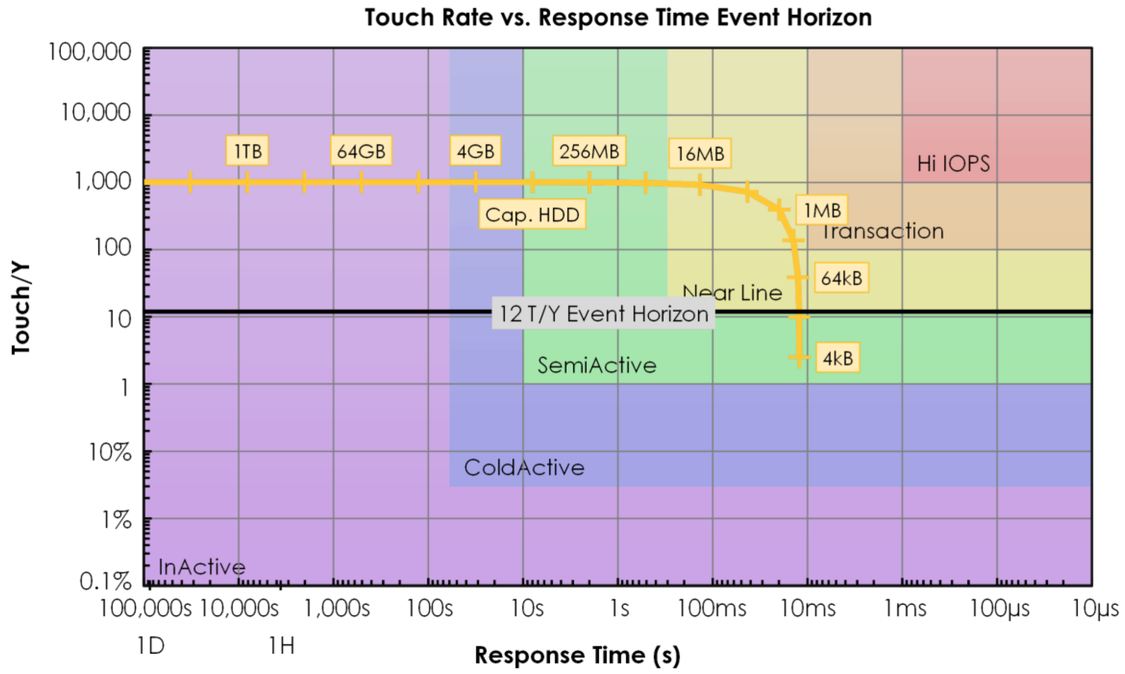
However, flash also suffers from a loss of touch rate with density (exacerbated by DWD limits). We thus anticipate similar trends for flash as the density increases. This will perhaps create an opening for other and faster storage technologies to displace flash memory in the future.

Archive Storage

It is important to have sufficient read touch rate to extract value from the data in a storage system during a given period of time. Data below the specified system touch rate is inaccessible – we call this being beyond the *data event horizon*. For example, if the touch rate falls below 1 for the time window chosen, some of the data on the storage device cannot be accessed within the window. **Figure 9** shows an example for a 4TB capacity HDD system where the application requires a touch rate of 12/Y.

In this chart we have an application that needs to touch the total capacity of the storage system in a month, which is a touch rate of 12/Y, and that we have chosen a system built from 4TB capacity HDDs. The require 12/Y can't be met with any object size less than 16kB. Thus, if the application needed a 4kB object size, some of the data would be untouchable in the month. Thus, we say operating with 4kB objects is beyond the data event horizon for this system. The data event horizon is a concern for many archive class systems.

Figure 9. Data event horizon for 4TB capacity HDDs



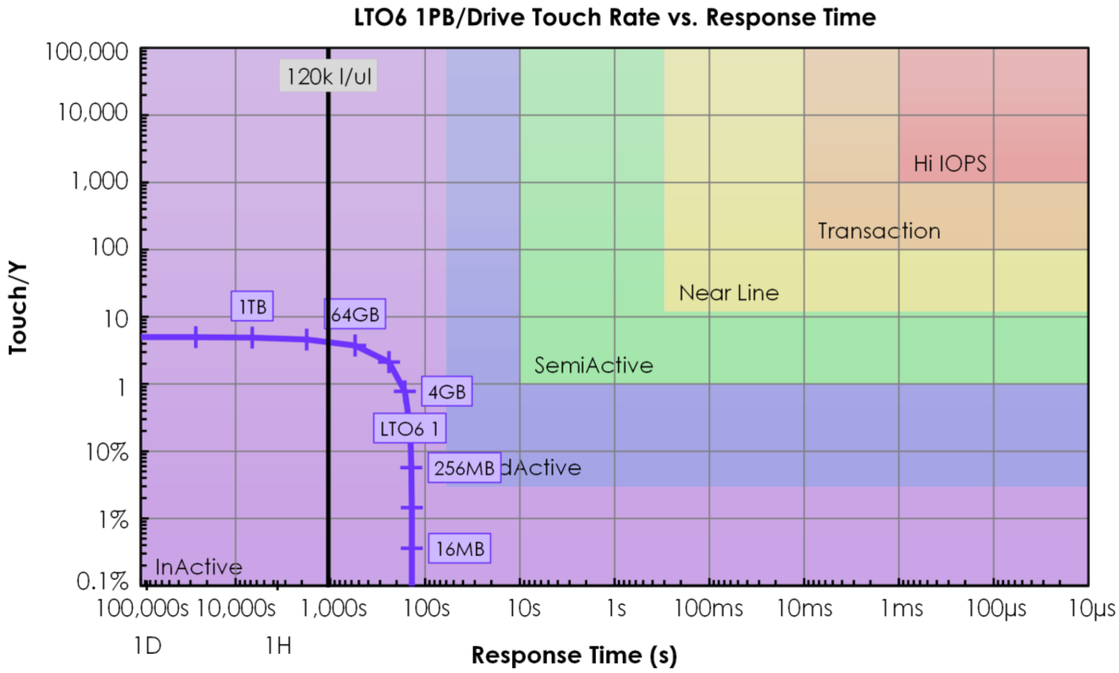
Tape

Tape has been widely deployed as an archive storage technology. **Figure 10** shows the touch rate curve for a magnetic tape system based on LTO 6 technology, with 1 PB of media capacity per drive. The ratio of media to drives influences the touch rate, so it's a key parameter for removable media systems. We assume that the system is not limited by robotic media delivery. LTO 6 media has 2.5TB raw capacity, so 1PB of media is 400 cartridges. We have picked this ratio as it is roughly where the media cost begins to dominate the system cost. So, adding more drives will increase the $/TB, but adding more tape won't lower the $/TB much. Thus, it is the highest touch rate which can be achieved at close to the media $/TB.

The small object response time here is just over 2 minutes (140s). The saturation touch rate is 5/Y. The tape curve is clearly situated in the InActive archive section with these characteristics. It should be noted that the touch rate for even moderately sized objects is very low. At a 16MB object size, the touch rate is only 0.4%/Y. The touch rate doesn't saturate until the block size is larger than 64GB. This explains why tape is used for large sequential transfers – where the object size is many GB.

We can see that this performance is getting near the edge of usefulness for archiving. Just filling the archive with data requires a touch rate of 1 for the fill interval. It might be argued that a cost effective way to deploy tape would be to purchase the needed capacity quarterly instead of annually, so it doesn't have too much unused media. This is a write touch rate of 1/quarter, or 4/Y. The read touch rate is then whatever the applications require, and may be less than 1/Y. For example, if only 10% of the data is needed in a year, we can have a read touch rate of 10%/Y. Thus, the total touch rate required is 4.1/Y.

Figure 10. Magnetic tape touch rate chart



The astute reader will have noticed that we over-engineered the system for the above example. We gave it a touch rate of 5/Y across the entire data capacity, whereas we only require it for ingest writing. For an InActive archive, the update rate for the data is very low (update includes delete operations). Thus the cost can be reduced by have a touch rate of 1/write interval, considering the portion of data deployed in that interval.

So, we need our touch rate of 1/quarter for 1 quarter's worth of data, which is a touch rate of 1/Y for a year's worth. Thus, we can write all of the data in a year even when deploying the data quarterly. So, the limit for tape is more like a touch rate close to 1/Y. You could in principle go a bit lower by considering the capacity for more than 1 year. The required write touch rate would drop proportionally.

However, we need to reserve some IO capability for actions such as scrubbing, drive cleaning and the like. This will increase the touch rate required. For example, a once per year full data scrub obviously requires a read touch rate of 1/Y. Thus, it seems unlikely that systems with a touch rate below 1/Y will be of significant value outside of some very InActive workloads.

Practical tape limits

Figure 10 showed the touch rate curve for a tape system considering only the drive and robotic system performance. Unfortunately, the curve shown can't be achieved in practice due to system constraints. As mentioned earlier, flash devices have performance restrictions that limit the touch rate. One restriction tape systems have is a limit on media full file passes (how many times the tape goes over the head), which places an ultimate limit on the touch rate. A full file pass is defined as reading the entire capacity of the tape. Once a media lifetime is chosen, we can easily convert this into a touch rate limit. For example, a full file pass limit of 300[4] and a 4-year media life (this might be too short for a long term archive) gives an ultimate limit of 75/Y. Longer media lifetimes will of course reduce the ultimate touch rate.

The drives themselves have limitations on the number of load/unload operations. For LTO6, this is typically 120,000 load/unload cycles.[5] A 4-year drive life this gives 30,000 load/unload operations per year. There are 8,760 hours in year, so this gives a max IO rate of 3.4 operations per hour, which is a minimum response time of 1,050 seconds. This limit line is labeled 120k l/ul in **Figure 10**. (The spec in reference 5 is claims a 5 year drive life, which would increase the minimum response time to 1,300s.)

We need to recompute the touch rate curve with this limit in place. The resulting achievable touch rate curve is shown in

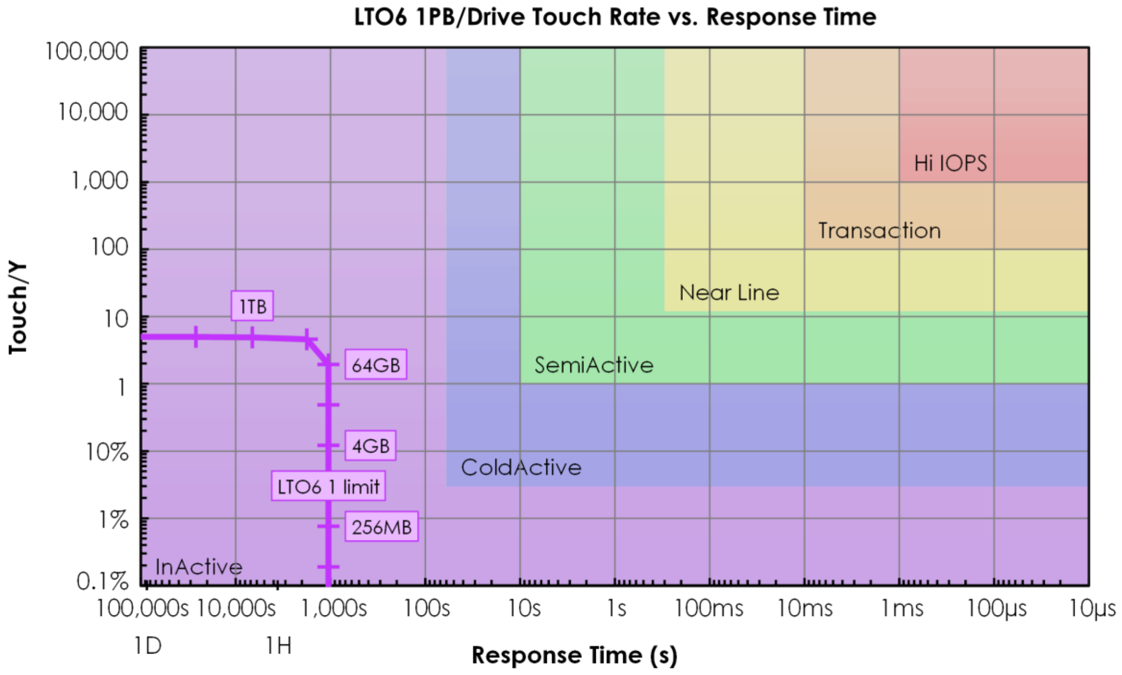
Figure 11. There are two impacts. First, the curve is compressed to the left, with a minimum response time of 1,050s. Second, the touch rates for the object sizes in the vertical portion of the curve have shifted down as well. In the unlimited curve, the touch rate for 16MB objects was 0.4%/Y, here it has dropped to 0.05%/Y. (Note we have treated the load and unload as a single operation against the limit.)

Enterprise tape systems can improve things somewhat, as they tend to have higher limits on load/unload cycles and faster data rates as well.

[4] IBM Tape Library Guide for Open Systems Redbook, Feb. 2015, p152.

[5] HP LTO Ultrium 6 Tape Drives Technical Reference Manual Volume 4: Specifications Sep. 2012, p25.

Figure 11. Tape touch rate reflection drive lifetime limits



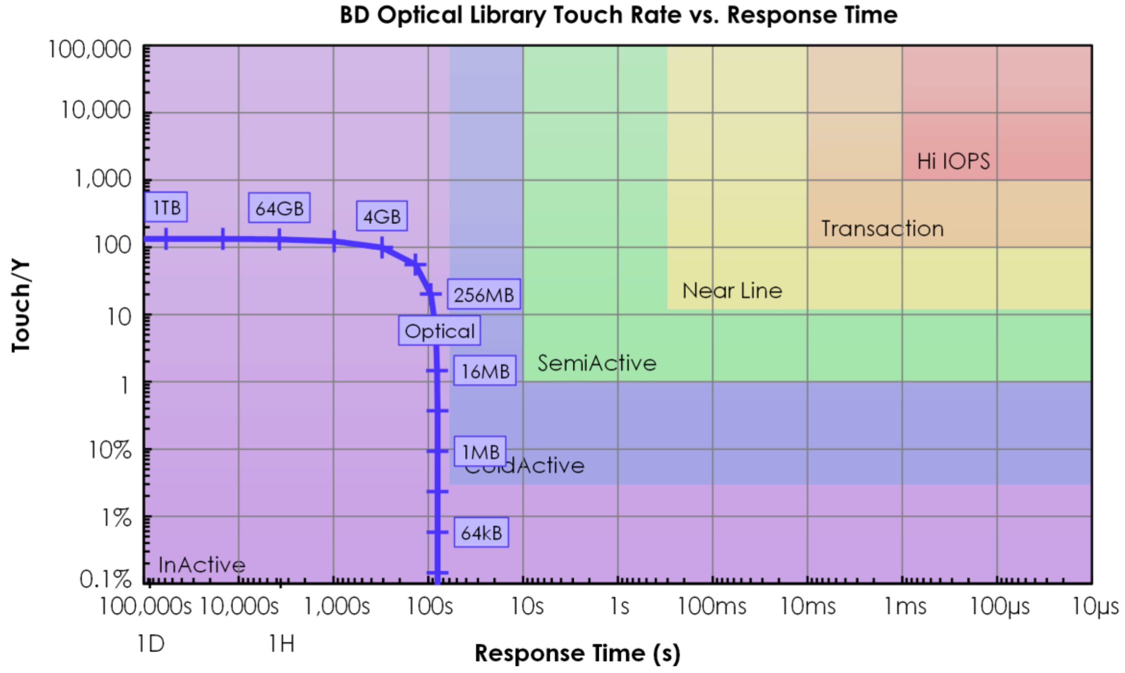
Optical

Once left for dead, optical is experiencing resurgence in the archive space. **Figure 12** shows the touch rate curve for a typical Blu-ray disc based optical library. This configuration has 4TB of media per optical drive. The minimum response time here is 80s. The saturation touch rate is 135/Y. The performance here is substantially faster than the LTO6 systems discussed earlier. This likely explains the increased interest in optical.

There are many different system configurations possible, which will have different performance characteristics. It is very likely that media cost will be a key factor in determining the success of optical systems. However, it is clear they can avoid the low touch rates of tape systems. The higher touch rate means somewhat higher cost can be tolerated as more value can be extracted from the data on optical.

At a 256MB object size, the optical system has a response time of 94s and a touch rate of 20/Y. Compare this with the LTO6 case, where the 256MB response time is 1,050s (11x slower), and the touch rate is 0.8%/Y (2,500x lower).

Figure 12. Optical disc touch rate chart



System Analysis Examples

We are now ready to use Touch Rate analysis to assist in storage system design. We will examine in detail three systems, a massive array of idle disks (MAID), Storage Tiering with HDDs and Tape in an archive and for flash memory tiered with HDDs.

MAID

Let's start by examining a massive array of idle disks, aka MAID. MAID systems have only a fraction of the disks powered on at a given time, trading response time for power savings. Thus, they are candidates for archive-class solutions. We can use touch rate to understand how such a system will behave.

First, the touch equation needs to be modified for MAID systems to account for the fraction of the drives active at a given time. This limits the effective data transfer rate for a given capacity. ActiveRatio is the fraction of drives which are active at one time. Thus, we now have **Equation 2** as an adjustment to the touch rate computation.

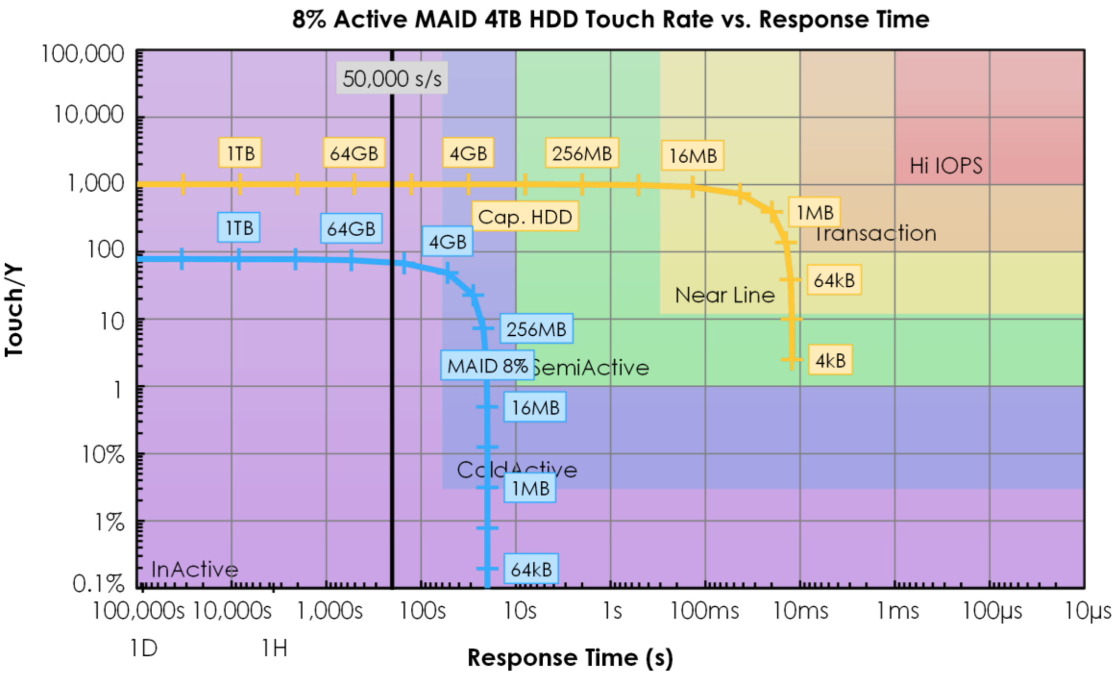
$$\boldsymbol{Touch/Y = \frac{ObjectSize(MB) \times 1000 \times ActiveRatio}{ResponseTime(s) \times Capacity(TB) \times 31.5}} \qquad \textbf{Equation 2}$$

An ActiveRatio of 1 will give the same results as **Equation 1** as expected.

The response time will also be adjusted to the activation time for the drives. If the idle drives are completely powered down in the MAID, then this is the time from power-off to active plus the object transfer time. The latency and seek time can be disregarded because the spin up time is 100x longer than either of these. For example, if 10% of the drives are active (an ActiveRatio of 0.1), then 90% of random IOs will require spinning up a drive. If we wish to model having a higher hit ratio to active drives, this can be adequately estimated by considering a larger object size.

If we build the MAID system using the same 4TB HDDs of **Figure 3,** we expect the touch rate to drop and the response time to increase significantly. The result is shown in **Figure 13** for an ActiveRatio of 0.08, compared with the 4TB capacity HDD.

Figure 13. MAID touch rate chart



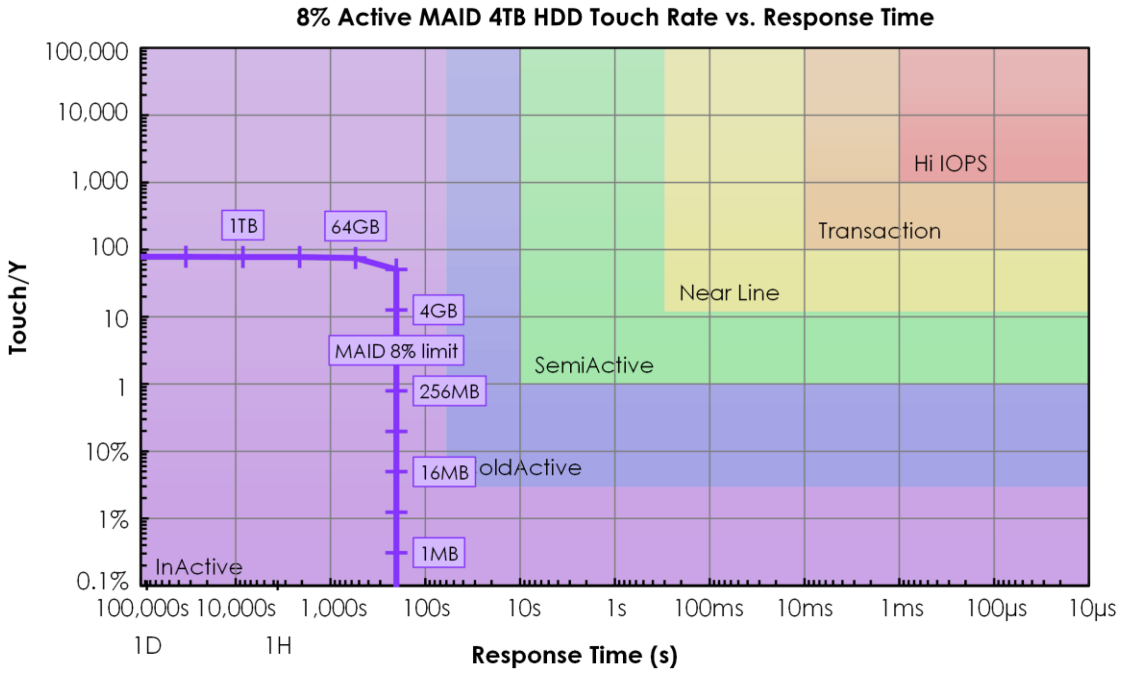
The 8% MAID system has a saturation touch rate of 80/Y, whereas the HDDs that comprise the MAID reaches 1,020/Y (the 8% MAID has 8% of the touch rate). The 4kB response for the MAID system is 20s, vs. 12.2ms for the HDD. The MAID touch rate at 4kB is 0.01%/Y, vs. 2.5/Y for the HDD. There is clearly a significant performance hit with 8% MAID. However, it will use only about 8% of the power of a fully active HDD system, thus the attraction for archives.

Unfortunately, there is more to the MAID story. Most capacity HDDs support only 50,000 start/stops.[6] If we want the drives to last for 4 years, this means a limit of 12,500 start/stops per drive per year. Now, each drive will spin up for only 8% of the IOs, so the system activity limit per drive per year is 12,500/.08 = 156,000. The minimum response

[6] HGST MegaScale DC 4000.B Hard disk drive specification Rev. 1, Oct 16, 2013 p30.

time is thus 200 seconds. This limit is shown as the 50,000 s/s line in Figure 13. The touch rate curve needs to be recomputed with these limits, as we did for the tape system. The result is shown in **Figure 14**.

Figure 14. MAID touch rate chart reflecting start/stop limits



This new curve has the same saturation touch rate, but the small object response time and touch rate are impacted by the drive limits. The minimum response time of 200s is now somewhat slower than the optical system of **Figure 12**, but overall the curves are quite similar.

Tiering/Caching

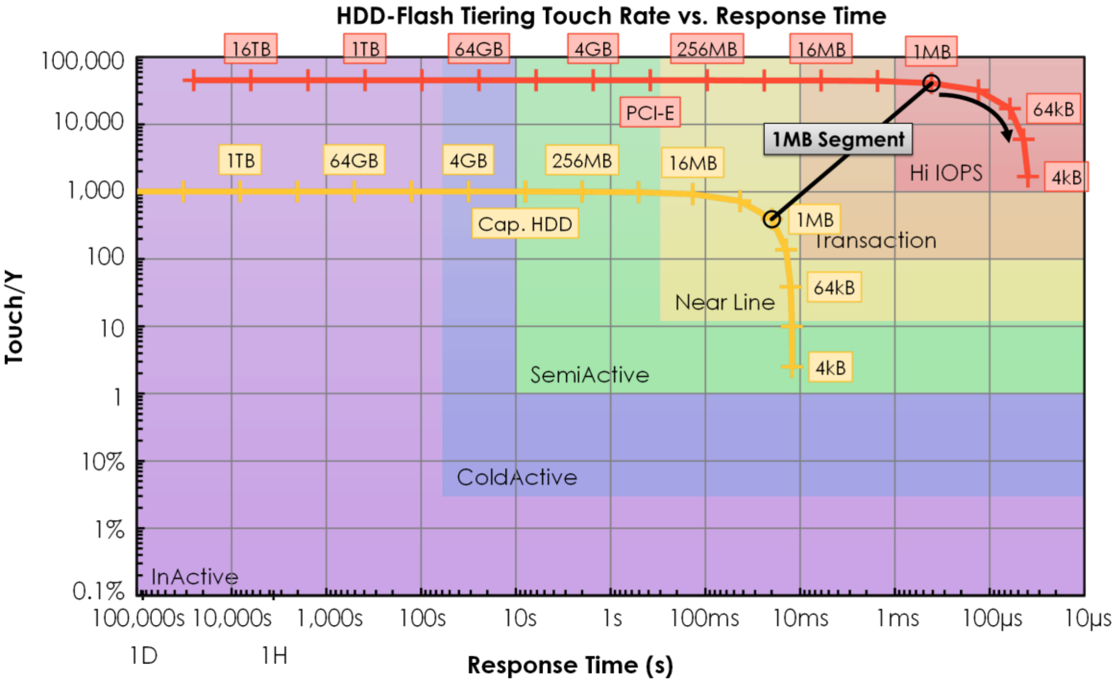
One method for improving system performance at low cost is to use caching or tiering to higher performance layers. From the standpoint of this analysis, the differences between caching and tiering are small. Caching tends to involve frequent data movement between the layers, and there is always capacity in the lower layer reserved for the data in the cache. Tiering tends to move data between the layers less frequently, and may not have reserved capacity in the lower layer (it exchanges data).

The touch rate chart can be used to understand the value extractable from either of these approaches. In either method, a data segment chunk size is chosen to move data between the layers. (In caches, this is commonly referred to as the line size.) The concept is that there will be multiple IO hits into the upper layer segment per data move. If this isn't the case, then the value of tiering will be low. Fortunately, most workloads do benefit from such data movement.

Figure 15 shows a touch rate chart for tiering HDD to PCIe flash. A 1MB segment will read random 1MB objects from the HDD layer, and transfer the data to the flash layer. This is shown as the black line in the figure. The time for the data copy is the slower of the two response times. Here it is the HDD response time of 20ms, giving an effective data transfer rate of 50MB/s. The data can then be operated on at a smaller object size on the flash layer.

Let's assume the application uses a 16kB object. The touch rate for 16kB objects on the flash layer is 6,100/Y, while it is only 10 for the capacity HDD. The tier can also access the entire HDD system capacity more than once per day, since the HDD touch rate for 1MB objects is 400/Y. Thus we can get a performance improvement if the hit ratio is high enough. We will cover the details of how to work out touch rate including the tiering/caching behavior in a future paper.

Figure 15. HDD-Flash tiering/caching touch rate chart

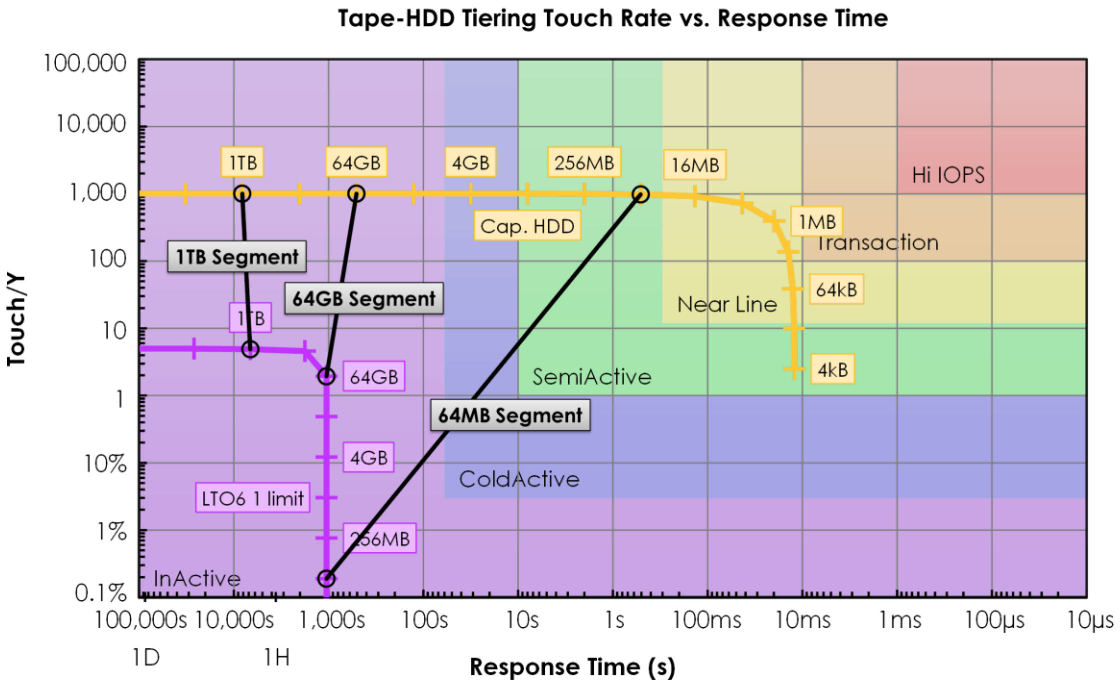


We can also look at the situation for tiering tape to HDD, shown in **Figure 16**. The situation here is more complex, given the slow response time and low touch rate for tape. Here, we highlight three segment sizes: 64MB, 64GB and 1TB. Using a 64MB segment will be of limited value, since the tape touch rate is 0.2%/Y. Thus, only 0.2% of the data in the tape layer can be moved to the HDD layer in a year. It is difficult to imagine situations where this would be adequate. However, the data transfer time is the same for a 64GB segment, where the touch rate is a more practical 2/Y. However, this means fewer segments are available on the HDD tier, thus the hit ratio will likely be adversely impacted.

The situation for a 1TB segment size is interesting since the HDD speed limits the transfer time.

Finally it should be pointed out that we haven't examined the full benefits or costs from tiering here. For example, the response time and touch rate will be dominated by IO misses, which require data movement from the lower layer. Thus, there is a practical limit to effectiveness of caching or tiering based on the touch rate difference between the two layers. For example, the touch rate of the tape-HDD tiering won't be improved noticeably by having a faster upper layer such as flash.

Figure 16. Tape-HDD tiering touch rate chart



Conclusion

We have introduced the concept of touch rate as scale-independent metric for measuring system performance. It also is a measure of the value that can be extracted from a data set on a storage system. In effect, it is a measure of the data inventory turns that can be achieved.

The metric is based on a few simple system parameters, making it suitable analyzing and comparing system designs. The object curve can be used to determine expected performance for different workload classes on different technology. The touch rate plot is thus a useful tool for quickly analyzing system performance.

We introduced the concept of the data event horizon to illustrate that it is possible to have data stored in a system that can't be accessed in a given time interval.

Touch rate can be used for widely varying system designs, including removable media and MAID.

We have given a brief introduction to application of touch rate to cached and tiered systems. A fuller treatment will be the subject of a forthcoming study.

Appendix: Touch rate terminology

Touch rate terminology

access_in (s) = the time from the IO request to the first byte of data (classical access time) in seconds. May be different for read and write operations.

access_limit = limit on the number of access events in the life of the component. For example, this could be a start/stop or load/unload limit.

access_out (s) = the time from the last byte of data until the system is ready for a subsequent IO in seconds. This is zero for SSDs and HDDs, but not for removable media systems such as tape. For tape, we need to include rewinding, unloading and returning the cartridge. However, this parameter could be used to model the impact of non-data IO operations such as garbage collection in SSDs and SMR HDDs. This parameter may be different for read and write operations.

ActiveRatio = fraction of the devices in a system which active at given time. Typically 1, but lower for system designs such as MAID.

Data event horizon = touch rate target limit for an application/system. Applications using object sizes below the data event horizon will not be able to access all the data in the given time window.

DWD limit = drive writes per day, which is a common flash device write limit specification. This is directly a write touch per day limit.

Object_MB = object size for the application in MB. This will typically be an average, but it's often worthwhile to examine the behavior at the actual application object sizes.

response_time (s) = the time from the IO request until all the data is received in a busy system in seconds (back-to-back IOs). Includes time waiting for the system to restore to an IO ready state. If queuing is used, this should include the queue wait time.

Unit TB = the capacity of a unit in TB. For most systems, this is the capacity of the unit that delivers the transfer rate (e.g. a hard disk). For removable media systems, it is the media capacity per drive.

xfer_limit = data transfer limit in TB/Y. Such limits are becoming more common for capacity hard drives, although we are unaware of anything physical that wears out with data transferred.

xfer_rate (MB/s) = the sustained data transfer rate in MB/s. May be different for read and write operations.

Computing Response Time and Touch Rate

The response time and touch rate can be computed from a few simple parameters for most devices and systems.

The response time in seconds is computed as:

$$response\_time(object\_MB) = access\_in + access\_out + \frac{object\_MB}{xfer\_rate}$$

In some circumstances, there may be a minimum response time.

The touch rate per year is computed as:

$$Touch/Y\ (object\_MB) = \frac{object\_MB \times 1000 \times ActiveRatio}{response\_time(object\_MB) \times unit\_TB \times 31.5}$$

The 31.5 is a scale factor for converting the units from per second to per year, given the capacity scale units. In some systems, there may be a maximum touch rate.

For a hard drive, we would have:

access_in = latency + mean_seek

acess_out = 0

xfer_rate = sustained data transfer rate

ActiveRatio = 1

Unit_TB = HDD capacity in TB

object_MB = appropriate object size for your workload

Touch rate spread sheet

We have created a basic spreadsheet with the math detail so anyone can work out touch rate for a particular system. The instructions are included in the spreadsheet. The spreadsheet can be downloaded at http://smorgastor.drhetzler.com/library or http://www.tomcoughlin.com/techpapers.htm.

A word on queuing

Queuing effects are not covered here as they generally require a more detailed performance model. In general though, queuing would increase the touch rate for a given object size, but the queue depth would also increase the response time.

About the Authors

Steve Hetzler

Steve Hetzler is an IBM Fellow at IBM's Almaden Research Center (San Jose, Calif.), where he manages Cloud Data Architecture Research. He has spent over 25 years in data storage research and development.

Dr. Hetzler is currently developing highly reliable, low cost storage system architectures for cloud like applications, and novel storage systems for tackling the big data explosion. He developed Chasm Analysis, a methodology for analyzing market potential for storage technologies using economic data. Previously, he initiated work on the IP storage protocol that is now known as iSCSI, which he later named. The group under his management developed the concept from an idea to the first specification before joining with Cisco to bring the work to the Internet Engineering Task Force. His team developed the first working iSCSI demonstrations. Steve has a blog on data storage at http://smorgastor.drhetzler.com.

Steve has been issued over 60 patents for inventions covering a wide range of topics including data storage systems and architecture, optics, error correction coding and power management. His most notable patents include split-data field recording and the No-ID(TM) headerless sector format, which have been used by all magnetic hard-disk-drive manufacturers for a number of years. He obtained his Ph.D. in Applied Physics at the California Institute of Technology.

Tom Coughlin

Tom Coughlin is a respected storage analyst and consultant. He has over 30 years in the data storage industry with multiple engineering and management positions at high profile companies.

Dr. Coughlin has many publications and six patents to his credit. Tom is also the author of Digital Storage in Consumer Electronics: The Essential Guide, which was published by Newnes Press. Coughlin Associates provides market and technology analysis as well as Data Storage Technical Consulting services. Tom publishes the *Digital Storage Technology Newsletter*, *The Media and Entertainment Storage Report*, and other industry reports

Tom is active with SMPTE, SNIA, the IEEE and other professional organizations. Tom is the founder and organizer of the Annual Storage Visions Conference (www.storagevisions.com), a partner to the International Consumer Electronics Show, as well as the Creative Storage Conference (www.creativestorage.org). He is the general chairman of the annual Flash Memory Summit. He is a Senior member of the IEEE, Leader in the Gerson Lehrman Group Councils of Advisors and a member of the Consultants Network of Silicon Valley (CNSV). For more information on Tom Coughlin and his publications. go to www.tomcoughlin.com.