Storage Switzerland, LLC

Visual SSD Readiness Guide

Prepared by: George Crump, Senior Analyst
Prepared on: 6/2/2009

Ready for Solid State Disk?

For many enterprises, Solid State Disks (SSD) should no longer be a future consideration. Their environment can take advantage of SSD right now and the organization can benefit from the inherent performance improvements that SSDs offer, such as increased revenue generation, customer satisfaction, and employee productivity. The jump to SSD is not a shot in the dark; users can make an informed decision by utilizing common utilities in the data center and a little guidance on what to look for.

Prior to running any tests or benchmarks, the first question to ask is: Are there applications in the environment that would produce greater benefit for the enterprise if their performance was improved? Types of benefits might include such things as orders processed more quickly, improved customer satisfaction, or customers not leaving your web site because they are tired of waiting (only takes a few seconds now days). Can your employees generate more digital work if they do not have to wait on the IT environment to respond or wait for a simulation job to run? If the answer to any of these questions is yes, then the time to look at SSD is now.

Unlike in the past when SSD price was an issue, it is no longer necessary to consider only applications that are extremely bottlenecked, nor do they have to be segment-able, meaning there are specific components of the application that could be moved to high speed SSD. The cost of SSDs is now such that even applications that could see a modest performance gain with a modest increase in productivity or revenue are viable candidates. These applications no longer have to be sophisticated databases where only specific files require moving to SSD. The size of the modern SSD is such that in some cases the entire application and all its data can be moved to the high-speed platform.

<u>These days, any application that is storage I/O bottlenecked and could benefit the organization if it performed better is an SSD candidate.</u>

For the purposes of this article we will use screenshots from Windows Perfmon to demonstrate the characteristics to examine when evaluating the viability of SSDs. We will show a series of before and after SSD performance results. For these Perfmons, nothing else changed in the environment other than the insertion of an SSD system.

These screenshots, provided by Texas Memory Systems, are based on real customer data; the customer in question is an online gaming platform with thousands of concurrent users. Each user or player pays to use the service, so game response time is critical for customer retention. A positive, responsive gaming experience leads to a customer that not only will continue to renew their membership but also refer other players to the service.

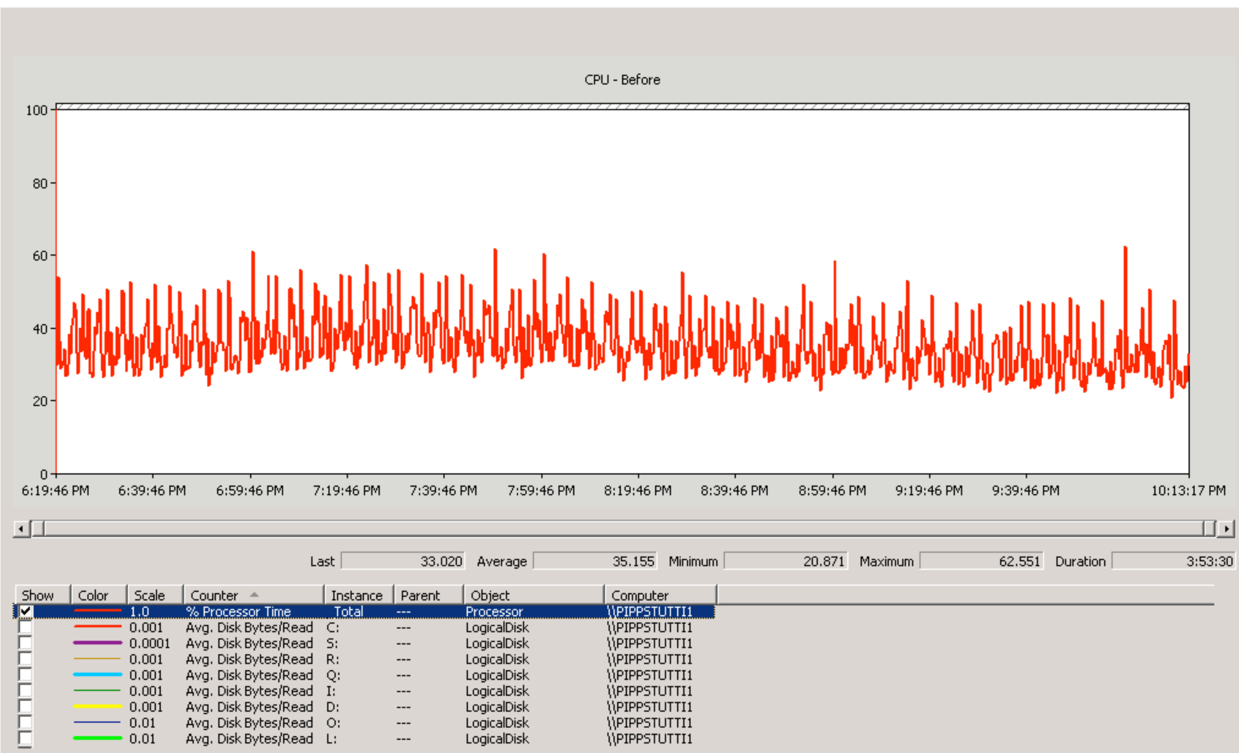
Do you have a storage performance problem?

Once a candidate application for SSD has been identified, the next step is to make sure the application is indeed storage I/O bound. First, look at processor utilization. There are three immediate areas that indicate a storage I/O problem: CPU utilization, bandwidth utilization, and disk I/O's per second (IOPS). The first and probably most telling step in this process is to examine CPU utilization.

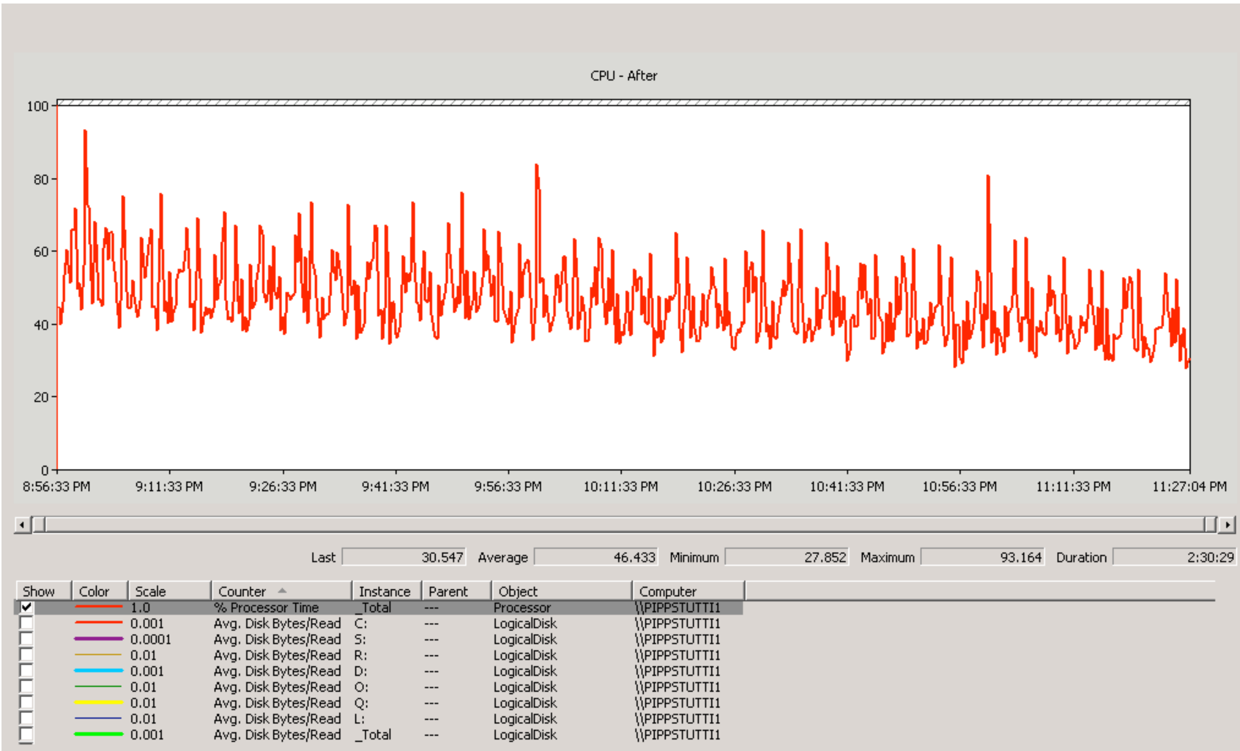
What we are looking for here is relatively low CPU utilization; below 50% indicates a problem, below 30% is not uncommon. If processor utilization is only 30%, that means CPUs are waiting on something else 70% of the time. Most often, they are waiting on storage.

If CPU utilization is already high, 60% or higher, then stop; it is unlikely at this point that there is a storage related performance bottleneck. If there are faster processing capabilities available, scaling up in compute power is the next logical course of action. After that, the decision to scale out by adding multiple servers to the application and possibly compartmentalizing it further across those servers should be carefully considered.

In the "before" Perfmon screen shot below you can see that the CPUs in the environment were not under a great deal of stress at all, with each averaging about 35%. Clearly this is a candidate for Solid State Disk.



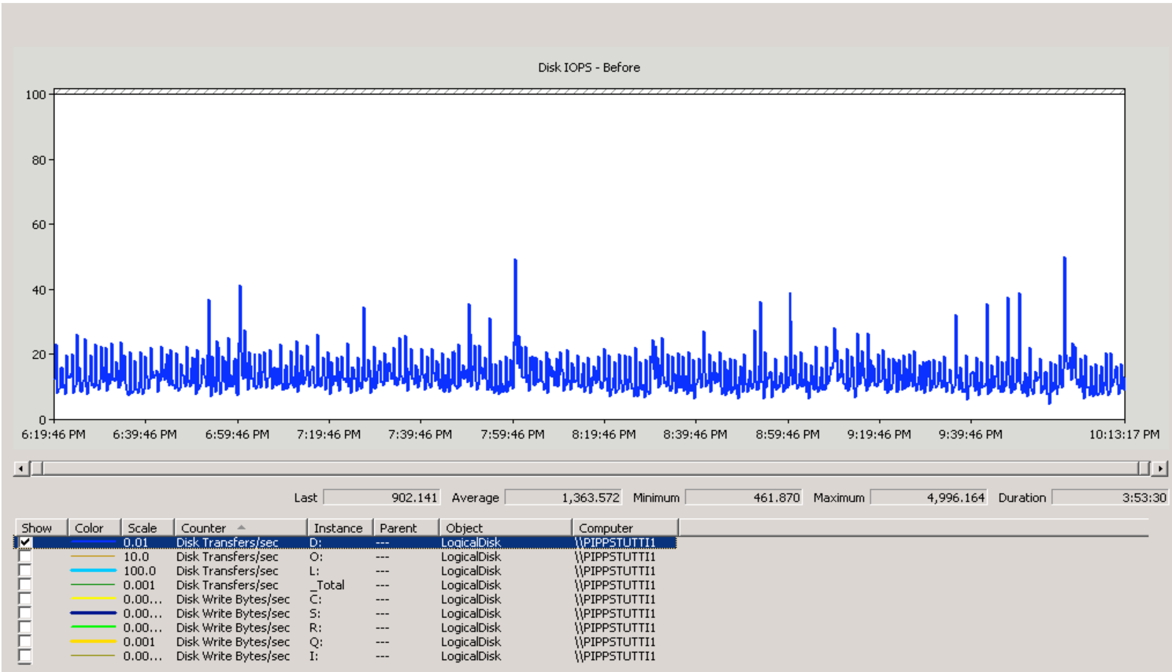
After the insertion of SSDs, there was a dramatic increase in CPU utilization to 46%, close to a 50% improvement in CPU utilization. With SSD you will find that I/O performance should scale linearly with CPU load.

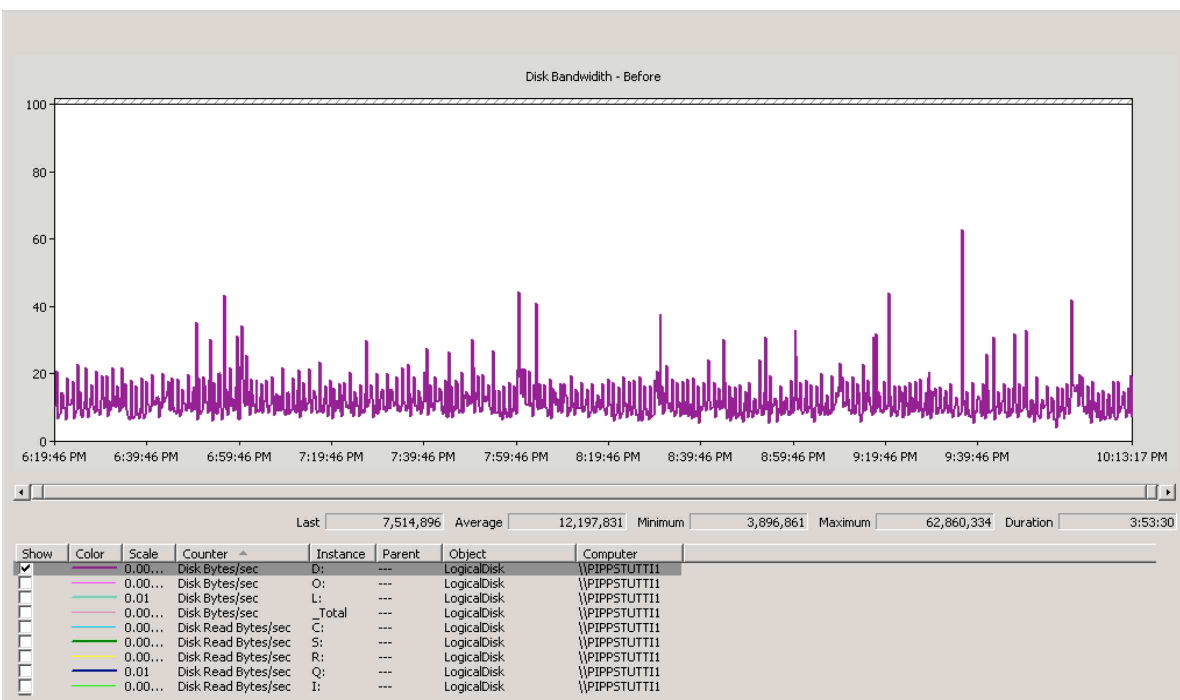


The goal of SSD is to make the CPU the bottleneck, essentially moving the performance problem back to the CPU. At that point there are two options: increase the processing capabilities of the environment by purchasing faster processors, or re-write the code to be more efficient in its use of the available processing power.

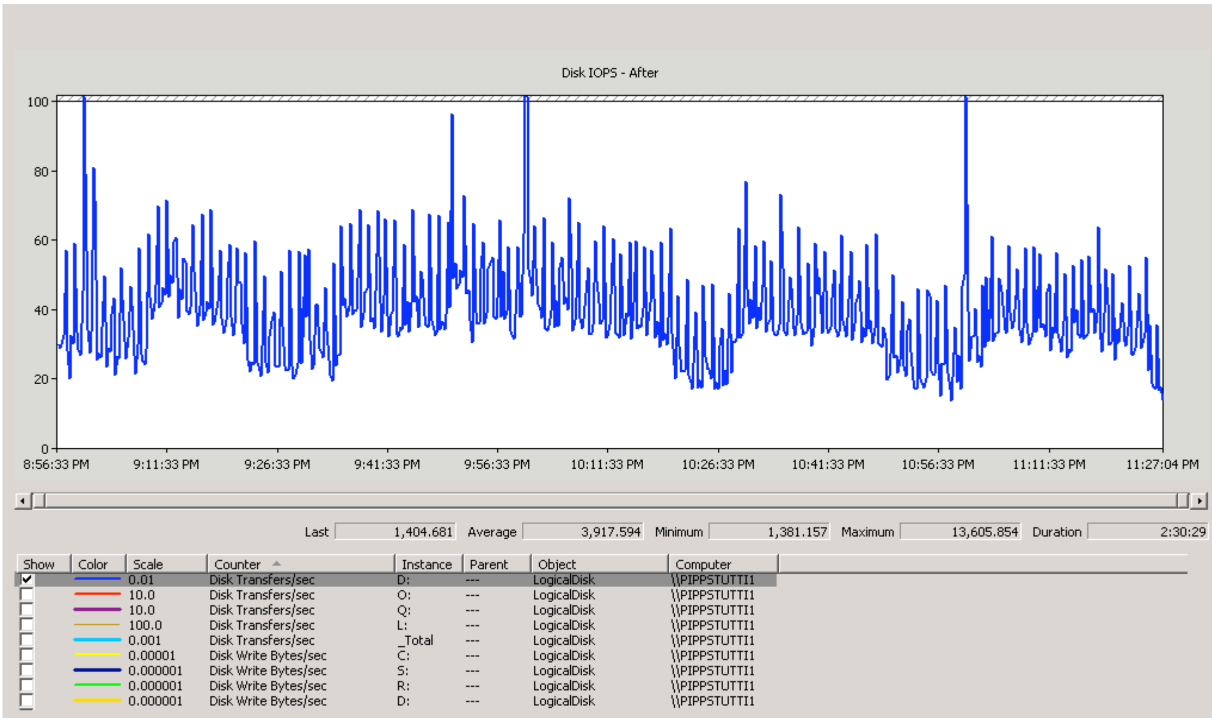
There are times, especially with a constant user load such as 1,000 gamers for example, where there may not be a dramatic increase in I/O performance after SSDs are deployed. But in this instance there will almost certainly be a dramatic decrease in response time.

In addition to CPU utilization, it also is advisable to look at disk IOPS and disk bandwidth. Notice below in the “before” disk bandwidth and disk IOPS Perfmons the near flatline of the measurement. This indicates that performance has hit the wall.





Notice in the “after” screenshots below that the measurements are more scattered. This indicates that there is plenty of available performance and bandwidth for the applications.



If after running these measurements and before implementing SSD you have low CPU utilization (the before CPU utilization screenshot) but scattered disk IOPS (as in the after Disk IOPS measurement), then it is probably due to a network issue. Basically the CPU is waiting on something but it is not storage.

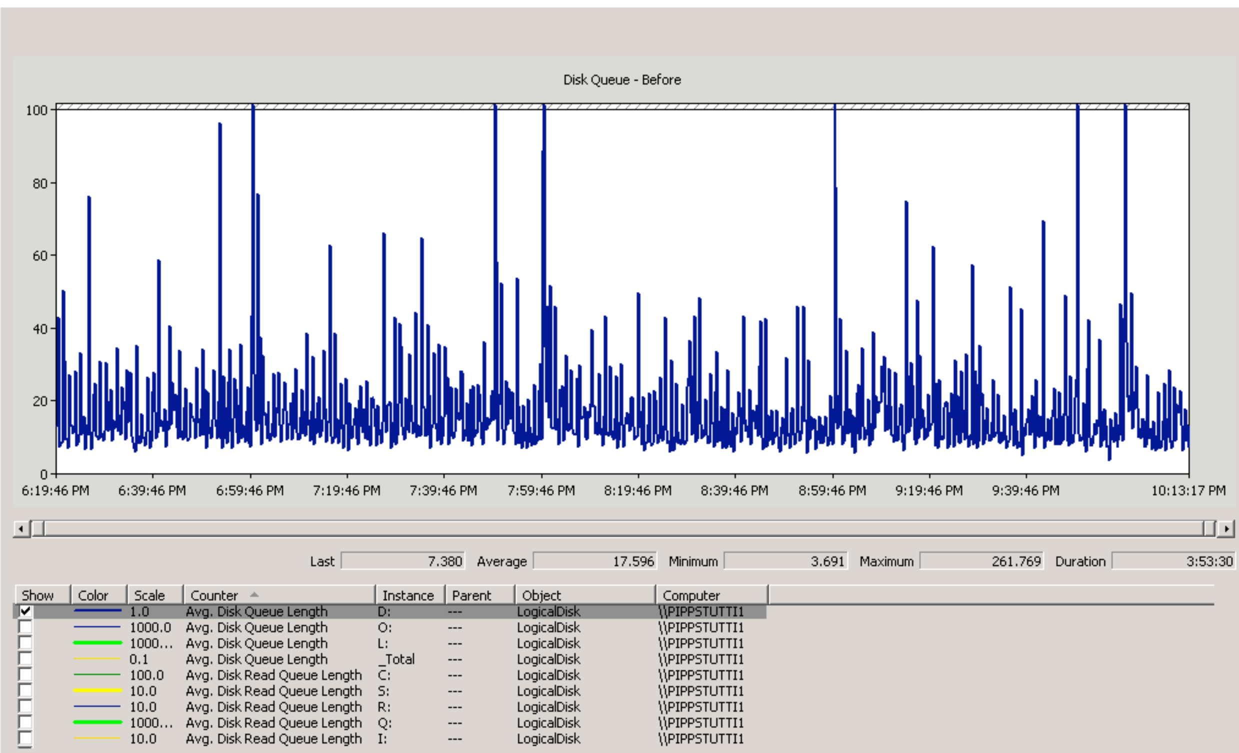
After identifying a performance problem, it makes sense to determine if you can address your storage I/O issue by improving your mechanical drive infrastructure or if it makes more sense to implement SSD.

Add more drives to the array or replace with SSD?

The first step most IT professionals will take in addressing storage I/O issues is to expand the number of drives in the array group. The problem with storage is that adding extra spindles to the array may not help. This is where the next Perfmon parameter to examine comes in: queue depth. Queue depth is the number of parallel commands that can be sent to a common target. Basically it is the number of requests that can be generated simultaneously by the application or users.

To saturate a RAID disk with 10 drives you must have a queue depth of 10; if you have a queue depth of 20 then the application could support 20 drives and so on. At some point it is likely the application will not be able to develop a deep enough queue to support a higher number drives, or the number of drives will become so high that it's no longer cost effective to use them. For example, 300 15k drives are likely more expensive than most SSDs available today.

The screenshot below shows an average queue depth of almost 18. That means that the application on average could keep about 18 drives busy and could peak to 262 drives occasionally.



For this reason, addressing queue depth is the storage equivalent of adding multiple processors or cores instead of one very fast processor. Like processors, this is often done because the next leap in single unit performance either is deemed too expensive or may be technically impossible with today's technology. At some point the bridges are crossed and the ability to scale up is once again preferable, so for a growing number of applications SSDs are a bridge ready to be crossed.

To some extent processors are a good example; it thus far has not been cost effective to scale up to an 8ghz processor. Instead there are multiple 3ghz cores per processor, essentially a scale out within the processor itself. In storage there was a scale up in RPM speeds from 5k to 10k to 15k. We have not yet seen 20k RPM drives and we may never see them. Engineering and economic issues as well as the rise of SSD systems are working against the technology. SSDs are the next scale up option in storage.

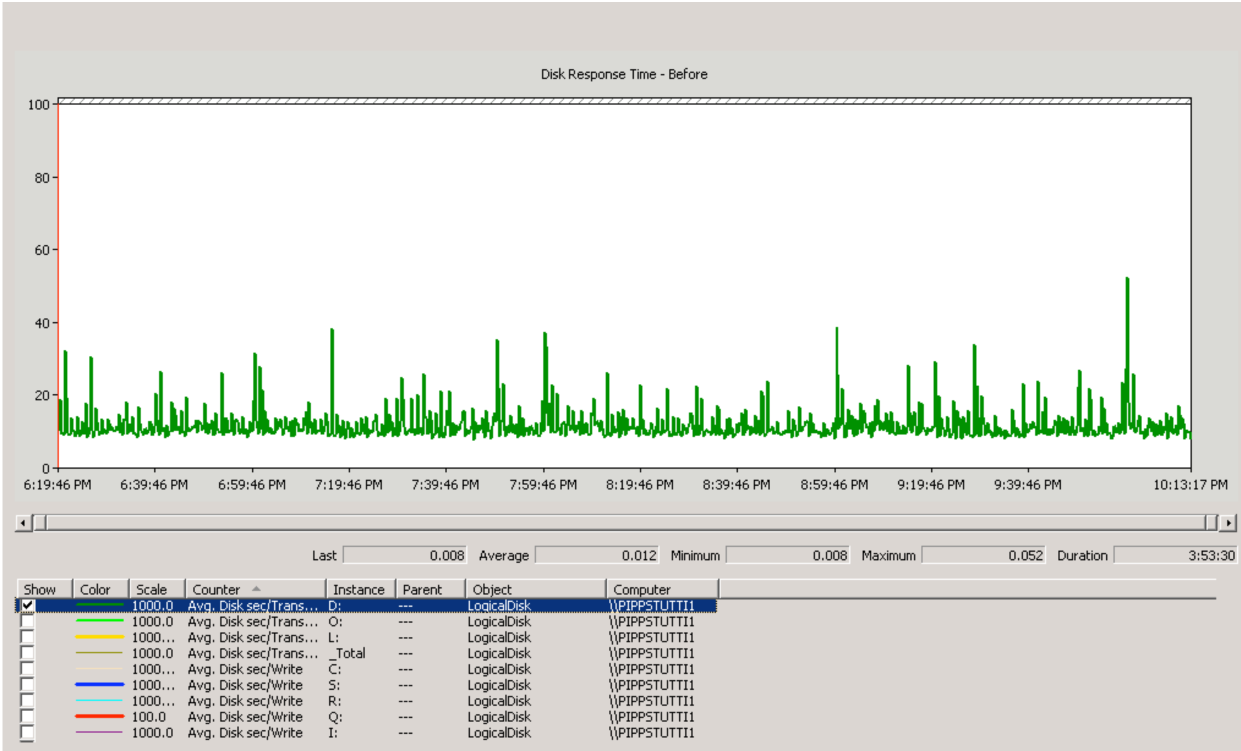
If it can be technically achieved, scale up is always preferred; it is typically less expensive when all the variables are considered. The cost of scaling out storage involves more than the expense of the drives; the cost of power, cooling, and maintaining those drives must also be included. If they can all be replaced by a single SSD, that could be less expensive out-right and especially so when power, cooling, and management costs are factored into the equation.

SSD's impact on queue depth has more to do with its response time, thus an explanation of response time is required before showing the SSD queue depth screenshot.

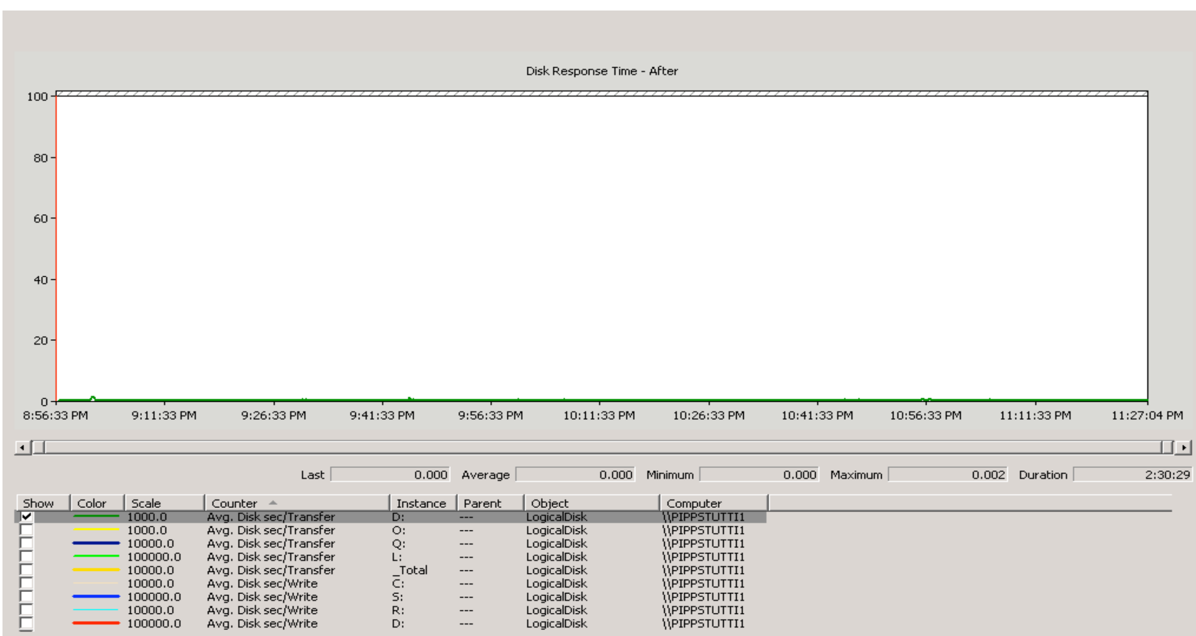
Eventually queue depth or scale out reaches a point of diminishing returns. In the case of storage, often sufficient queue depth cannot be generated to keep up with the number of drives available. The option at this point is to decrease response time or latency. Storage latency is the amount of time that the application or processor has to wait for the storage to respond. CPUs basically have two states; they are either processing something or they are waiting on something.

Response time then has to do with how fast the individual drive or RAID group as a whole can respond to a request. In hard disk storage this has to do with the speed of the drive, limited today to 15k RPM, so the next step is to place data at an optimal location on the drive, which is the outer edge. To ensure that this happens, some manufacturers recommend short stroking the drives to make sure the data is stored only at the fastest point on the disk. This of course leads to wasted capacity, increased power requirements, and having to explain to management why 3/4 of a system's capacity is not being used.

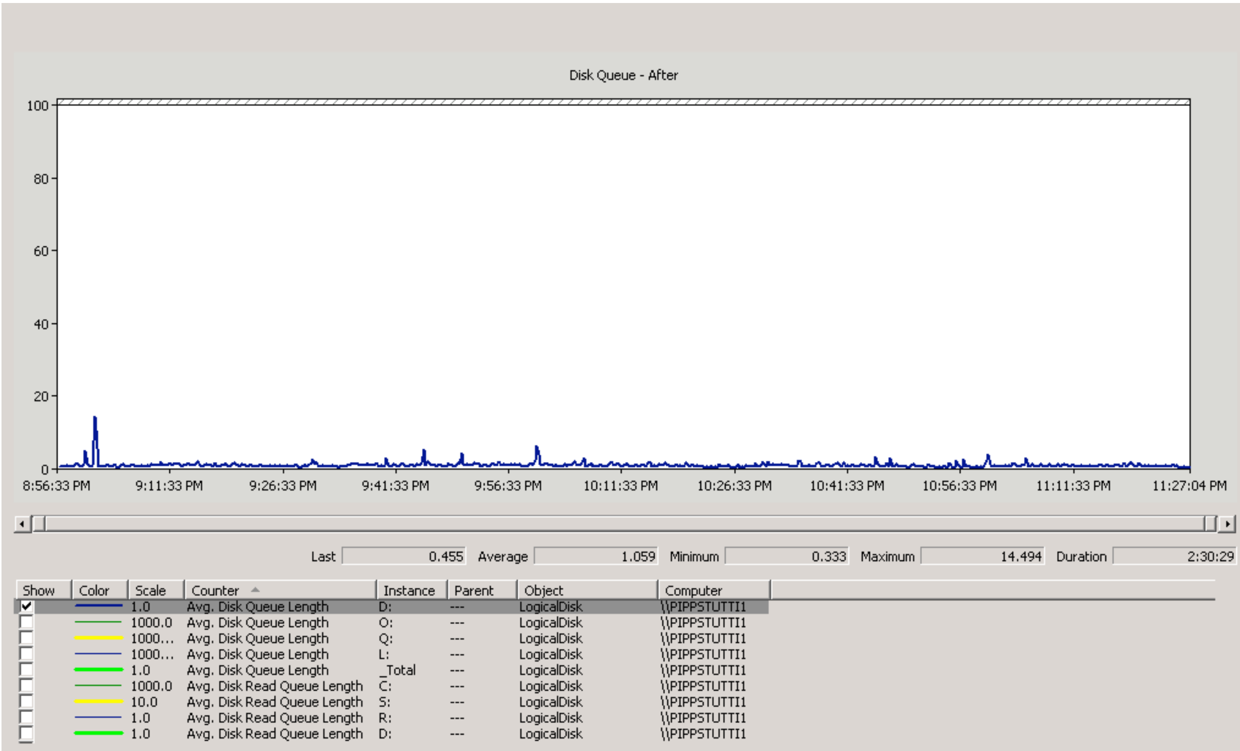
In the “before” chart below you can see that there is a measurable response time.



In the “after” chart you can see that response time, or latency, is all but eliminated. The primary impact on adding SSD to an environment is reducing response time.



This dramatic reduction in response time leads to a dramatic reduction in queue depth, not because the application can send more parallel requests but because its response from storage is so fast that it can't develop a very deep queue, as seen in the queue depth "after" screenshot.



These screenshots indicate that the customer saw a dramatic performance improvement from implementing SSD. They show that instead of scaling out by adding more drives and servers, users can scale up by adding SSD. As mentioned previously, scaling up is usually simpler and more cost effective than scaling out.

With solid state storage, the simple rule is that if speeding up the application will make more money for the organization or satisfy more users with less hardware, implement it. The big shift in the past 18 months has been that the cost to move to SSD has fallen dramatically. As we discussed in our "SSDs are Cost Effective Now" article, it is likely that SSDs are now less expensive than the work-around required by mechanical drives.