Executive Summary:
The advent of high-IOPS (I/Os Per Second) all flash arrays has revolutionized enterprise storage. Hundreds of thousands of IOPS are easily achieved in small, 3U boxes instead of racks and racks of disks.

Now executives looking for the utmost in application availability and consistent performance need to look beyond IOPS. Sustainable performance as exemplified by storage latency is the new I/O issue that system architects and planners need to confront.

Fortunately, many vendors have TPC-C and SPC-1 benchmarks that quantify latency. While all vendors now claim sub-millisecond response times, the fact is that almost all flash arrays have "long-tail" latencies in the dozens of seconds.

These long-tail latencies reduce server performance and increase system response time. They can also cause serious problems up the storage software stack. With each long-latency I/O the system has to decide if the problem is an I/O failure and what to do about it. A flash array with demonstrably low long-tail latency eliminates these problems, allowing application and system software to avoid long-latency corner cases, improving application availability and overall response times.
Introduction
The IOPS’s numbers of today’s flash arrays are formidable. Specifications of 500,000 to 1,000,000 IOPS are common. And unlike similarly spec’d disk arrays, the flash arrays take a fraction of a rack instead of multiple racks.

These numbers are so large that they end the problem of I/O sufficiency for all but the most demanding applications. But that does not mean our I/O problems are solved. The issue of latency in the remainder of the storage stack remains. The software stack of kernel drivers, file systems, databases and applications, to name a few, is far from optimized for a low-latency, high-IOPS world.

Latency reduction
Hard drives have been latency’s bottleneck for decades. Disks are so slow that the storage software stack’s latency contribution was ignored - a sensible strategy for decades, until now.

Researchers at the University of California found that with a 4Kbyte disk access, the standard Linux software stack accounted for just 0.3% of the latency and 0.4% of the energy consumption.

With a flash access however, the same software stack accounted for 70% of the latency and 87.7% of the energy consumed by a 4K I/O. The latency of the software stack is now a major issue.

Research also finds that the software stack isn’t nearly as efficient as it could be. For example, caching is enabled at virtually every layer of the stack. Yet caching creates overhead - table lookups for example - while also consuming resources by storing multiple copies of data.

Power implications
Since most of the software stack exists on the server, the power and resource implications of the server and/or storage controller need to be considered as well. Traditional approaches for reducing disk latency uses complex and resource intensive software to reduce disk accesses.

Yet servers and controllers are expensive, both in dollars and energy consumption. With high-performance flash storage, the old strategies are no longer cost effective - or needed.

Simply put, high-performance storage with lower power consumption is much more efficient than complex server-side software working to optimize what no longer needs optimization.

Low latency today
There is no quick fix for the inherent inefficiencies of today’s storage stack. In the longer term we have to re-architect the stack to optimize it for lower latency.
Until that re-architecting is complete - if it ever is - should we care about low latency storage now? Yes, because enterprises can take advantage of low-latency storage today.

**Benchmark data**

TPC-C and SPC-1 benchmarks using solid state storage are instructive. Many flash-based storage systems have long tail latencies into the dozens of seconds - despite average response times below 1ms.

Looking at flash-based system costs of under $1 million and with TPC-C throughput ratings over 1 million, the Violin arrays, running with either Cisco UCS or HP server hardware, achieved the lowest maximum latency of any tested configuration.

**What is the TPC-C benchmark?**

From the Transaction Processing Performance Council website:

*TPC-C simulates a complete computing environment where a population of users executes transactions against a database. The benchmark is centered around the principal activities (transactions) of an order-entry environment. . . . While the benchmark portrays the activity of a wholesale supplier, TPC-C is not limited to the activity of any particular business segment, but, rather represents any industry that must manage, sell, or distribute a product or service.*

In the first TPC-C benchmark, with Cisco’s UCS hardware and two Violin arrays, the longest transaction response time was 5.39 seconds. In the second TPC-C benchmark, the longest transaction response time was 3.97 seconds.

In contrast, other flash-based systems had significantly higher I/O maximums, some as high as 10 times the Violin-based systems.

That I/O variability is a serious problem. Application and operating system software stacks shouldn’t have to deal with it. Their impact on performance and reliability are two good reasons why.

**SPC-1 benchmark**

More recent data is sometimes available from the Storage Performance Council’s SPC-1 benchmark,
which only measures array response times, not the entire application stack as TPC-C does. The SPC says:

The SPC Benchmark-1 (SPC-1) is a sophisticated performance measurement workload for storage subsystems. The benchmark simulates the demands placed upon on-line, non-volatile storage in a typical server class computer system.

SPC-1 provides slightly less granularity in latency measurement, topping out at 30 milliseconds and beyond. Unlike TPC-C there is no specific maximum latency reported.

For example, in an early 2015 filing by a major storage company for their flagship flash storage array, the Executive Summary shows that the array’s average latency was under 1 millisecond.

However, in the Response Time Frequency Distribution Data we see what the average times don’t tell: millions of I/Os took over 5 milliseconds and hundreds took over 30 milliseconds. These long-tail times aren’t any better than average disk-based array response times.

Looking at recent flash TPC-C and SPC-1 benchmarks show the same pattern: low average times with surprisingly high long-tail response times. In contrast, the two TPC-C benchmarks using Violin’s 6000 series arrays demonstrate very short maximum times.

**Why are latencies so high?**

All flash arrays make heavy use of parallelism and concurrency for performance. This is especially important for flash-based systems. Individual flash chips have low bandwidth and slow writes, which requires parallelism to achieve performance.

For example, one flash issue is free space recovery or, as it is better known, garbage collection. As data is deleted within a block, current data must be periodically rewritten to a new block to recover the deleted data capacity. However, flash must write entire blocks (128KB and larger) at one time, which takes tens of milliseconds, much longer than a typical disk access.
The flash translation layer (FTL) in SSD controllers hides this, which means array controllers above an SSD don’t know when a critical block is rewriting. However, in the Violin architecture the array controllers know the status of each block. Blocks in the recovery process are allowed to complete while the requested block is reconstructed from parity data. No waiting for a read due to garbage collection.

Flash is a relatively new technology. Engineers are still working with flash manufacturers to optimize flash feature sets for enterprise storage. Violin’s demonstrated latency advantage reflects their extension of the principles of parallelism and concurrency beyond SSDs all the way to the individual flash chip.

**Importance of latency reduction**

Reduced latency enables servers to do much more work with the same CPU cycles and cache capacity. The more in-flight I/Os a server has to juggle, the faster its on-chip caches fill up. And when they do, server performance hits a wall.

With fast, low-latency storage, the number of in-flight I/Os is kept to a minimum. There are no "long-tail" I/Os to absorb system resources and slow response times.

Data caches are another issue. Meant to keep disk I/O to a minimum, caches add complexity and overhead at every level of the stack. Caching has a place - it just isn’t every place.

**The human perspective**

The human perspective matters as well. People tend to remember - and judge - based on their worst experiences. Unpredictable response times or the occasional long wait time blemishes otherwise exemplary service.

**The Engineering challenge**

More importantly, long latencies create problems that are difficult to reliably engineer around. When an I/O takes tens of seconds to complete, is it a storage failure? A network problem? What now?

Typically, architects of the different levels of the stack make different choices about when to declare an I/O failure. For example, disk drives intended for RAID arrays will declare failure in a few seconds rather than retrying for a minute or more. This "fail-fast" strategy enables the rest of the stack to adapt quickly.

Each layer of the stack has its own code for dealing with I/O failures. A common problem with these error handling code paths is that they are not well-exercised. If the average latency is 900 milliseconds and the 90th percentile latency is 4 seconds, the
less common 10 to 50 second latency code paths won’t be used very often. If the code has an occasional bug, who will figure that out?

But if that 10-50 second latency never occurs the issue never arises. As practitioners of Statistical Process Control know, reducing process variability is a critical success factor to achieve high quality. Low latency/sustained IOPS storage will reduce information processing variability as well.

**Conclusion**

Flash has revolutionized enterprise storage. Past concerns about configuring disks to achieve high performance - short stroke disks; meticulous data placement; stripe sizes; heavily cached controllers - are much less important today.

But not all flash arrays are created equal. Some arrays have significantly lower long-tail latency than others - a number that is now more important than raw IOPS.

Re-architecting the storage stack for much lower latency is a long term project. In the meantime, architects and system planners should pay careful attention to the long-tail latencies reported in the TPC-C and SPC-1 benchmarks.

Unfortunately, some major vendors choose to not benchmark their systems. With these vendors buyers are essentially flying blind.

Buyers are encouraged to use these benchmarks to better understand how great the latency variation can be among similarly priced and specified systems. There are important differences hiding behind the impressive specs of today’s flash arrays.

---

**About The Author**

Robin Harris is the president and chief analyst of TechnoQWAN LLC, publisher of StorageMojo.com. He has over 30 years experience in the IT industry in product management and marketing, business development and strategic planning at companies large and small. He earned degrees from Yale University and the Wharton School of the University of Pennsylvania. Robin lives in northern Arizona and may be reached at robin [at] StorageMojo.com