

What do IOPs and Bandwidth have in common?

In an ever evolving world of networking, storage and computing in general, we are being bombarded by information which is meant to lead us towards finding the “best” product or feature that are out there. Vendors are often parading infographics and statistics in front of us to give us the “obvious” benefits of their product versus their competitors, or our own existing infrastructure.



Now I’m not saying that the statistics that are being used aren’t meaningful, but we also have to be able to have a good understanding of what it is that we are using in a comparative analysis.

Understanding Measured Values

Numbers quoted as fact can’t be denied. But the important factor with what we use with measured values is that we are using something that gives is the proverbial apples to apples comparison. Many times we are presented with statistics and values that aren’t fair comparisons at all.

Take for example the pitch about cream cheese, which is touted as having “Half the calories of butter”. This is an absolutely true statement. But let’s be honest here:



Have you ever put half an inch of butter on a bagel? The real measurement that we have to use is percentage of daily intake by serving size, and then ensure that we use the serving size correctly.

This is an easy trap to get caught in when you are being groomed for a purchase of a product and the smiling vendor or consultant is pointing at some charts and graphs that clearly put their performance ahead of others.

So What do Bandwidth and IOPs have in common?

There are two often quoted measurements for product performance which are **bandwidth** and **IOPs (I/O Operations per second)**. Both are very valid units of measurement for a product or topology. We have to be absolutely aware of what our performance is with % of bandwidth utilization and stats like peak IOPs, average IOPs.

What both of these measurements have in common is that they are only one part of the overall picture. And truthfully, they aren’t really the most important part.

The Elephant in the room: Latency

In both networking and storage, the real key measurement that we have to be mindful of with our designs is really latency. Latency is the measure of time (usually in milliseconds) that it takes for data to get from point A to point B.

Latency is the difference between synchronous and asynchronous transactions. Latency is what will cause delays from data passing between portions of your systems that could result in application

timeouts and failures.

There isn't a single product out there that can rid us of latency. Do you know why? Let's just ask this guy:



We don't have to have be Einstein to understand latency luckily, but we do have to know that it is a matter of physics. No amount of bandwidth can eliminate latency.

Bandwidth versus Latency

Just like with storage, your network is often measured by the "size of the pipe" which is the bandwidth allocation. I have a 100 Mbps connection between site A and site B which seems like more than enough. But when I dig into the numbers a little more, I will quickly find that there is more to the picture than just the bandwidth.

FACT 1: My sites are 5000 km apart. Nothing can be done to change this. But luckily networks use optical transmission which means that we our data travels at the speed of light!! Oh wait...

FACT 2: Light travels at 299792 kilometers per second. So we have our first limitation. Right out of the gate we have our baseline of best possible latency:

LATENCY (in milliseconds) = $1000 * \text{distance} / 299792$

So in my example, the very lowest latency for the round trip is 16ms. Right out the of the gate I have some delay which then will be amplified by many other factors such as jitter, line quality and hop count which each lead to nominal but real delays in the speed of the transfer of data.

Beyond that, we also have to account for our packet sizes. Remember that we aren't just shipping a 25MB file in a big chunk from site A to site B. Each data packet (assume 1500 byte MTU for our example) has the data portion of the packet and the header. From this we know that our file is now a number of TCP packets that require send time and acknowledgement time (remember that TCP is connection oriented) so the bandwidth is being further broken down by the amount of blocks we can send.

DATA SPEED (bps) = $8 * \text{block size} / (\text{distance}/299792 + 8 * \text{block size} / \text{bandwidth})$

So for our example of a 100 Mbps connection, with 5000 km of distance and a 1500 byte MTU, we end up with a data speed of 714362 bps. Ouch.

So when we talk about replication and DR/BCP configurations, you have to understand the limitations of latency in your physical topology

Demystifying the "We can do 200,000 IOPs" statement

This is a great claim to have. And it is absolutely true. But we have to look at the other statistics that come into play which is the latency of the transactions, and also the type of data that is being sent.

The ability to serve up 200,000 IOPs, but of 512 byte data packets, is not what we need to measure. So we look at how they can handle larger data. How about 4KB? I bet that you can still serve up 200,000 IOPs (note: probably reads, not writes) but at what latency?

If the read or write has 60ms of latency, it most likely will cause a timeout or delay in the handling of the I/O by the source system. Especially if it is a high performance database application which generally requires < 10 ms read times and < 5ms write times.

You have the wrong kind of data

When we are given statistics for IOPs and compression by storage vendors, we have to be acutely aware that the numbers being presented have been generated in near-perfect conditions. What that means is that the data being used for generating high volume I/O reads and writes is specifically formatted.

Sequential and random reads and writes will change the graph entirely when we measure storage performance, and this is important. Not only do we have to know what the pattern of our data movement is, but we also have other downstream effects.

If an application develops latent write response, it will delay in-application transactions while the writes are committed. This becomes a compounding effect as the rest of the application may stack up I/O to be fed to the storage.

So when we have proposed data speeds of X and real-life data speeds of Y when we roll out to our own environment, we are often told that we have the “wrong kind of data” because it doesn’t work well with the product.

What does this mean?

All that I wanted to note with this is that there are a number of factors that are involved in understanding storage and network performance and your application design. When you are staring at charts and graphs from a vendor, you have to be able to ask questions like “What type of data is being used? What is the nature of the reads and writes? What is the average versus the peak? What are the hard limitations of distance and latency on your performance?”

Always ask about what the peak latency will be, and what the effect of IOPs will have on average latency for writes as load is applied to the system. Ask about what reads and writes are done in cache, and what the actual write to target will be in ms so that you can fully understand the real lifetime of the transaction.

You will be able to get real information from vendors, but we have to be sure to ask the right questions and we have to be aware of the type of data we have which will be on these systems, and the nature of how the data is managed and consumed.

Don’t get lost in the shiny infographics. Remember, that this is the only pie chart that is absolutely true:

