Introducing a New Performance Metric for Scale-Out Storage Systems

This whitepaper introduces a simple and intuitive new storage performance metric, with particular applicability to scale-out storage systems. There is a pressing need for such a performance metric, given the rise of public and private clouds and of scale-out data centers.

We begin by describing what applications expect from storage and how block storage systems are architected to satisfy application requirements. Then, we describe some traditional storage metrics that are used to compare storage systems, and we explain why each one is flawed in some way. Finally, we introduce a new performance metric which we call the Performance Efficiency Percentage (PEP). While we are particularly interested in scale-out storage systems, this metric is equally applicable to scale-up storage systems.

Application Requirements from Storage

What does an application need from storage?

- It needs to be able to store a certain amount of data (GBs).
- It needs to do a certain number of I/O operations per second (IOPS) against that data.
- It expects a certain latency when accessing that data.
- It often needs data protection from up to 3 types of failures.
  - Component or subsystem failures such as storage media, power supply, storage processors, and so on. Protection from this type of failure is often expected from the storage system.
  - Catastrophic failures such as an entire data center failure. Protection from this type of failure is sometimes provided by the storage system, but higher layers of software will often provide this function. Protection typically involves synchronous or asynchronous replication, at distance.
  - User or application error. Protection is typically provided by backup software.

Applications may be running on bare-metal or they may be running in VMs or containers. If they run inside VMs on top of hypervisors, the desired IOPS must be delivered to the VM and must account for any I/O virtualization overhead in the hypervisor that can take away from the performance seen by the VM.

Background on Storage Systems

Storage systems can be scale-up or scale-out. For this discussion, let us focus on block storage systems which expose volumes (or LUNs) to applications.

About the Author

Jai Menon

Jai is Chief Scientist at Fungible. He joined Fungible after having served as CTO for multi-billion dollar systems businesses at both IBM and Dell.

Jai was an IBM Fellow, IBM's highest technical honor, and one of the early pioneers who helped create the RAID technology which is behind what is now a $20B durable storage industry. He impacted every significant IBM storage product between 1990 & 2010, and he co-invented one of the earliest RAID-6 codes in the industry called EVENODD. He was also the leader of the team that created the industry's first, and still the most successful, storage virtualization product. His team at IBM also built one of the fastest and earliest parallel file systems in the world. When he left IBM, Jai was CTO for the IBM Systems Group responsible for guiding 15,000 developers. In 2012, he served as VP and CTO for Dell Enterprise Solutions Group. In 2013, he became Head of Research and CRO for Dell.

Jai holds 53 patents, has published 82 papers, and is a contributing author to three books on database and storage systems. He is an IEEE Fellow and an IBM Master Inventor, a Distinguished Alumnus of both Indian Institute of Technology, Madras and Ohio State University, and a recipient of the IEEE Wallace McDowell Award and the IEEE Reynold B. Johnson Information Systems Award.

The volumes may be raw (no RAID or erasure coding) or durable.

Scale-up storage systems typically include two (sometimes a few more) processing complexes and some number of storage devices. Let us assume the storage devices are SSDs for this whitepaper. The scale-up storage system is designed to be highly available with no single point of failure (it has two or more processing complexes that can take over for each other, RAID or erasure coding to protect from SSD failures, dual power supplies, and so on). To get higher performance, sometimes the storage system will allow the processing complexes to be upgraded to more powerful processors with more cores or more memory. This upgrade can sometimes be done non-disruptively. But ultimately, the performance cannot increase beyond what the most powerful processing complex supported by the storage system can deliver. To increase capacity, a scale-up system may allow the addition of one or more additional drawers of SSDs, and/or it may allow an upgrade to larger capacity SSDs. Ultimately the maximum capacity supported is topped out when the largest capacity SSDs are used and the maximum number of drawers supported is deployed.

Scale-out storage systems, on the other hand, typically deploy a cluster of storage nodes, each with a single processing complex and some number of SSDs. An entry scale-out system that supports durable volumes will deploy a small number (e.g. 2 or 3) of storage nodes to ensure that there are no single points of failure. The nodes are network connected to each other. Recovery from failures of both SSDs and storage nodes is achieved by using RAID or erasure coding across nodes. Since nodes can be in different racks, scale-out systems can be more durable than scale-up systems. Furthermore, scale-out systems can achieve higher performance and/or higher capacity by adding more storage nodes to the cluster. The aggregate performance or the aggregate capacity of the cluster is only limited by the maximum number of storage nodes that can be supported in a cluster. Any given volume can be spread out across multiple storage nodes in a cluster, and the storage cluster appears as a single large storage system to the applications.

Both scale-up and scale-out storage systems may provide features such as RAID, erasure coding, compression, deduplication, thin provisioning, snapshots, clones and encryption.

Traditional Metrics of Storage Performance

Let us look at some traditional metrics of storage system performance and discuss the shortcomings of each one.

In this paper, we focus on throughput oriented metrics where we perceive a greater need for improvement. For latency, metrics such as zero-load latency, average latency, and tail latency[1] do an adequate job; hence we will not discuss latency further in this whitepaper.

IOPS A traditional metric reported by all storage systems vendors is the IOPS provided by their storage system. There are two problems with this metric.

First, IOPS is very workload dependent, so the quoted IOPS may or may not be appropriate for your workload. As an example, if the storage system uses a read cache, the quoted IOPS may assume a particular hit ratio which may be different than what your workload might achieve. Second, an application or customer cannot decide if a storage system is fast enough for their needs from just looking at the published IOPS metric alone.

To understand why this is the case, consider two storage systems that both publish they can do 10M IOPS. See Figure 1. Consider also that they both use state-of-the-art SSDs with 4 TB of capacity, each capable of delivering an intrinsic maximum performance capability of 1M IOPS. Let's assume that vendor A's storage system can do 10M IOPS, but needs 250 SSDs (1 PB of storage) in order to achieve the stated IOPS. The 250 SSDs in the storage system have an intrinsic maximum performance capability of 250M IOPS, yet vendor A's storage system is only able to achieve 10M IOPS[2]. On the other hand, assume that vendor B's storage system can achieve 10M IOPS using only 15 SSDs (60 TB of data and 15M IOPS of intrinsic maximum performance capability). Vendor B's product is clearly superior and can satisfy more demanding application requirements, at a lower cost, than vendor A's product. For example, an application that needs 10M IOPS on 60 TB of application data can be satisfied by vendor B using 15 SSDs, whereas vendor A will need 250 SSDs to satisfy the same request.

1 Tail latency is often expressed as the 98th or 99th percentile latency.
2 This may be due to the high overhead of providing storage services such as compression, encryption, deduplication and so on. It may also be due to limitations of the interfaces connecting the storage system to the data center.

This means if the customer chooses vendor A, they will need to purchase 1 PB (250 x 4 TB) of total capacity, even though they only have 60 TB worth of application data to store. Storage systems provide many features such as RAID, erasure coding, compression or deduplication, encryption, snapshots and the like.

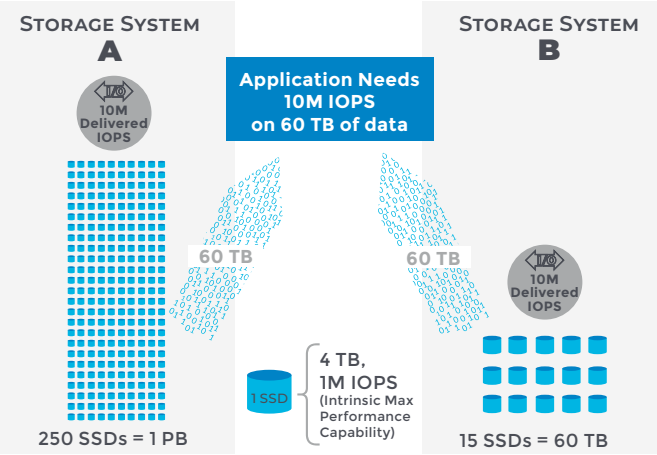


Figure 1. Comparing storage systems with different delivered IOPS

The reason that storage systems cannot deliver the inherent capability of the underlying SSDs is because these features can impact performance[3]. Another reason could be because the storage system has limited network bandwidth to the application servers.

IOPS/GB Since applications specify both their IOPS and their storage requirements (GBs), it would be helpful for storage systems to publish the IOPS/GB they are able to deliver. This is a better metric than IOPS alone[4], as it allows an application to decide if a given storage system can better meet its needs from this single metric.

Scale-up storage systems have traditionally not published such a metric. However, increasingly, scale-out systems are beginning to do so. For example, AWS Elastic Block Storage (EBS) has two SSD-based offerings – one that publishes a 3 IOPS/GB capability and another that publishes a 50 IOPS/GB capability.

IOPS/Eff-GB This is a refinement of the IOPS/GB metric. Customers and applications cannot use all the raw available GBs or capacity in a storage system, due to various overheads. We define the effective GBs (eff-GB) delivered by a storage system as the customer usable capacity – the actual number of GBs available to the applications. As an example, consider a storage system which provides 4+2 (4 data, 2 parity) protection for all stored data. In this case, eff-GB = 2/3 of available GB, since the storage system uses 1/3 of the available capacity for its own needs. If the storage system also uses a log-structured approach and needs to reserve 25% of space for garbage collection, then the eff-GB is further reduced. In this case eff-GB becomes 2/3 x 0.75 = 0.5 of available GB. On the other hand, if the system does compression and deduplication and on average is able to reduce storage needs by a factor of 5, eff-GB becomes 0.5 x 5 = 2.5 of available GB. A storage system with twenty five 4 TB drives has an available capacity of 100 TB, but an effective capacity of 250 TB using the number in the above example. Eff-GB is then 250,000 GB. See Figure 2 for an example of how effective GB changes when different data services are enabled.

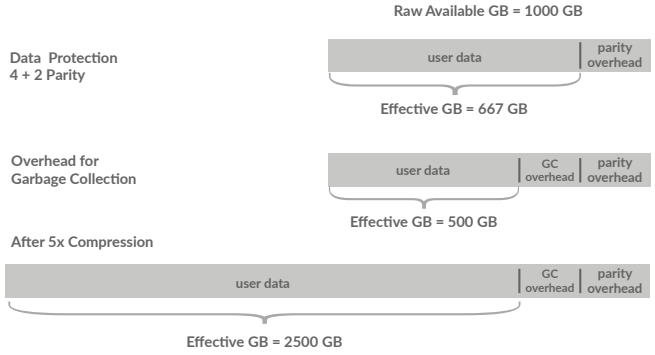


Figure 2. Changes to Effective GB when Different Data Services are Enabled

IOPS/eff-GB is a better metric for comparing two storage systems than either IOPS or IOPS/GB. However, there are several ways in which even this metric can still be gamed.

One way is by using lower capacity SSDs. For example, if a storage system has 12 SSD slots available, higher IOPS/eff-GB can be achieved using twelve 4 TB SSDs versus using twelve 15 TB SSDs. Another way to game this metric is by using a read DRAM cache in the storage system which will improve IOPS/eff-GB for cache-friendly workloads.

3 To be fair, comparisons should only be made between storage systems that deliver equivalent functionality.

4 However, the shortcoming of IOPS -- that it may not be relevant to your workload -- extends to IOPS/GB as well.

$/eff-GB Since all the ways to game the IOPS/eff-GB metric will cost money to implement, one way to achieve fairer comparisons between storage systems is to have them publish their price ($/eff-GB) for the delivered IOPS/eff-GB. If the storage system offers different services with different IOPS/GB, then it will likely charge different prices for the different IOPS/GB offered. A storage service that provides higher IOPS/GB can charge more. Here are two examples:

- Amazon Web Services (AWS). The standard AWS SSD EBS offering costs $0.10 per GB-month and supports up to 3 IOPS/GB. The AWS provisioned IOPS offering, which supports up to 50 IOPS/GB, costs $0.125 per GB-month AND an additional $0.065 per IOPS-month.
- Google Cloud Platform (GCP) charges $0.040 per GB-month for their standard disk-based offering which supports 0.75 IOPS/GB and they charge $0.17 per GB-month for their SSD persistent disk offering which supports 30 IOPS/GB. The price/IOPS/GB for standard disk is therefore, $0.0533 and for SSD disk is $0.0057.

When storage systems or storage offerings provide both IOPS/eff-GB and $/eff-GB, we are now in a better position to compare storage systems or offerings. However, looking at the example above, it is still a judgment call as to whether 3 IOPS/GB at 10c/GB (AWS) is better than 30 IOPS/GB at 17c/GB (GCP). If my requirement is 3 IOPS/GB or less, the AWS offering is superior. If my need is closer to 30 IOPS/GB, GCP is the superior offering.

This approach of using the combination of two different metrics to compare storage systems is the best approach, however, it also has some shortcomings.

Summary – Comparing Storage Systems
To summarize this section on traditional metrics, all the current ways to compare the performance of storage systems have flaws. Given the metrics we have discussed and defined above, people currently compare storage systems using one of the following two approaches.

Approach A. One way to compare storage systems capabilities is by looking at what IOPS/Eff-GB they can deliver, for a similar set of storage services. Storage services include read caching in memory (DRAM or Optane), durability, encryption, QoS, compression, and so on. For example, it is fair to compare the best IOPS/eff-GB vendor A is capable of delivering for raw volumes (and no additional services) versus the same metric for vendor B. It is important to make sure that if one vendor is using read caching to quote their numbers, so is the other vendor. The challenge comes when one vendor uses 15 TB SSDs and another uses 4 TB SSDs to improve IOPS/eff-GB. This is why this approach can be flawed. Furthermore, this approach is flawed because it ignores costs.

Approach B. An even better way to compare storage systems is using $/eff-GB for similar IOPS/eff-GB. Comparing what vendor A charges to deliver 3 IOPS/GB to what vendor B charges to deliver the same 3 IOPS/GB would be fair; comparing it to what vendor B charges to deliver 50 IOPS/GB would not. Such comparisons are hard to make when different vendors do not offer similar IOPS/GB. As we pointed out before, it is still a judgment call as to whether 3 IOPS/GB at 10c/GB (AWS) is better than 30 IOPS/GB at 17c/GB (GCP).

A New Storage Metric – Performance Efficiency Percentage

In this section, we present a new performance metric to compare storage systems, particularly scale-out storage systems. Our objectives for the performance metric are:

- It needs to be intuitive to understand and simple to measure.
- It needs to apply at cloud scale.
- A storage system with a superior value for the metric should provide superior performance on a broad spectrum of workloads.
- The metric should not be affected by the size of the SSDs used, unlike IOPS/eff-GB.
- A single metric should suffice to compare storage systems without the need for judgment calls, unlike the approach of comparing $/eff-GB for comparable IOPS/eff-GB.

We describe our new metric in stages. We begin by showing how it could be applied to a single storage system or a single node of a scale-out storage system. Then, we show how to extend the metric to the entire scale-out storage system.

We call our new performance metric the **Performance Efficiency Percentage (PEP)**.

PEP for a Single Storage System or Single Node of a Scale-out Storage System

The PEP is calculated as follows and can never be larger than 100%. The numerator is the total IOPS[5] delivered by the storage system on a purely sequential workload which cannot benefit from caching of any sort. The denominator is the number of SSDs in the storage system multiplied by the IOPS capability of each SSD. This denominator is really the intrinsic maximum performance capability of the SSDs. A storage system could not possibly deliver any more performance than the SSDs are capable of, for such cache-unfriendly workloads. Intuitively, it is easy to see that our metric is simply a measure of how close the storage system comes to exposing all of the potential back-end capability of the SSDs to the requesting applications. We show this metric diagrammatically in Figure 3.

We argue that PEP is a generally useful measure of storage systems' capability, independent of workload. Though the PEP is measured using a cache-unfriendly workload, we show that the higher the PEP, the better the storage system will perform on **all workloads**.

It should be clear that storage systems which have a higher value for PEP will do better for cache-unfriendly workloads (such as sequential read, media streaming, workloads with no locality in space or time, and so on).

Now, consider cache-friendly workloads – say a workload with a read cache hit ratio of 80%. 20% of the workload's accesses still require access to the SSDs. Let us compare a storage system with 90% PEP to another with 50% PEP for this workload. Let us assume the SSDs are capable of 20M IOPS. The storage system with 90% PEP can deliver 18M (20M*0.9) IOPS on read misses, and it could, in the best case, deliver as much 90M IOPS aggregate to the applications (18M * [1/0.2]), assuming there are no other bottlenecks which limit it to a lower number. Using similar reasoning, the storage system with 50% PEP can only deliver 10M IOPS on read misses and will be limited to delivering 50M IOPS (10M * [1/0.2]) to the applications. We argue that, all else being equal, and in the absence of other bottlenecks[6], a storage system with better PEP should be capable of delivering better performance, even for cache-friendly workloads[7].

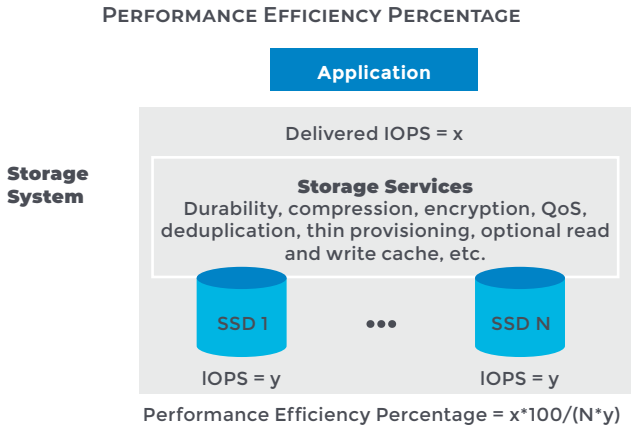


Figure 3: A New Metric for Storage Systems

To summarize, the better the PEP of a storage system, the better the performance it can deliver to applications, independent of workload. This is what makes it a broadly useful metric for comparing storage systems, in our opinion[8].

A higher PEP will always translate to better (or equal at worst) application performance for sequential or cache-unfriendly workloads. For cache-friendly workloads, a better PEP may not always translate to significantly higher performance due to other bottlenecks in the storage system.

The $/PEP Metric

To be fair, the PEP for two storage systems should be compared when delivering similar storage services. As an example, it would be unfair to compare the

5 It goes without saying that we could use GB/s instead of IOPS to create a related version of this metric.

6 Such as limited network bandwidth between the storage system and the application servers.

7 A storage system with higher PEP can never have worse performance on any workload. The worst-case scenario is for a workload with 100% hit ratio. In this case, the higher PEP storage system will perform as well as the lower PEP storage system, but not any better, since the SSDs are never accessed.

8 Storage systems may need to publish the value of this metric separately for reads and writes and for raw volumes.

PEP of a storage system which does compression and deduplication with another one that does not have these features.

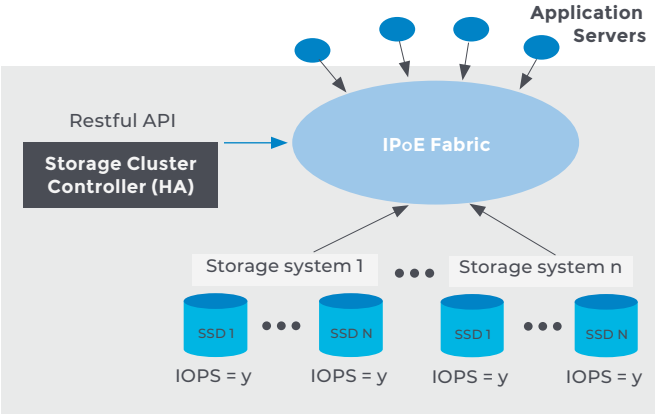
If two storage systems have similar PEP for a similar set of services, the lower cost one is clearly superior. One way to normalize for cost is to use a $/PEP metric to compare storage systems. We give two examples to illustrate the importance of $/PEP. Example 1: One storage system uses 20 processors to deliver PEP that is comparable to the PEP of another storage system which only uses 2 processors. Example 2: When 2 storage systems have similar read cache hit ratios, it cannot be assumed that the two cache designs are similar in cost just because they are equally efficient. One vendor may take $1,000 to build a cache that delivers the 80% hit ratio of the earlier example, and another might take $10,000 to deliver the same cache hit ratio.

For these reasons, we believe that $/PEP for similar services would be an important metric to compare storage systems.

PEP for Scale-out Storage

We discussed the PEP in the context of a single scale-up storage system or a single node of a scale-up storage system. We now discuss how to generalize the PEP so it applies to all nodes of a scale-out storage system in a data center. In this case, the denominator is the combined capability of all SSDs in all storage nodes in the data center. The numerator is the total aggregate IOPS delivered by all storage systems in the data center, across the fabric, to the applications running on the application servers, for a sequential or cache-unfriendly workload. This is shown diagrammatically in Figure 4.

PEP Applied at Data Center Granularity



PEP for Block Service includes

- all storage systems and associated control plane
- the fabric connecting app servers to storage systems
- NICs on app servers
- hypervisors on app servers when apps run in VMs

Figure 4. PEP Applied at Data Center Granularity

Conclusion

In this paper, we have presented a new performance metric to compare storage systems, particularly scale-out storage systems, called PEP. PEP is defined as

$$\text{PEP} = \frac{\text{(Delivered IOPS for sequential workloads)}}{\text{(\# of SSDs x SSD IOPS capability)}}$$

In our opinion PEP

- Is intuitive to understand.
- Is simple to measure. We can use the FIO tool and run a sequential workload to measure the numerator. The denominator can be computed from the number and type of SSDs in the storage system (from storage system specifications) and the number of IOPS each SSD can do (from the SSD manufacturer's website). Can be measured at the granularity of an individual storage node or at cloud scale.
- Is applicable to all workloads – both cache-friendly and cache-unfriendly ones, though it is measured by running a cache-unfriendly workload.
- Is independent of the number of GBs per SSD, unlike IOPS/GB.
- Can be extended to $/PEP as a way to compare the cost efficiency of storage systems. This is superior to the current approach of comparing $/eff-GB for a given IOPS/eff-GB, which requires making judgment calls.