



A SOLIDFIRE PAPER

# Guaranteeing Storage Performance

Everything you need to know about storage  
Quality of Service in a next generation data center

# Why read this guide?

“Managing and troubleshooting storage performance is a labor-intensive process burdening today’s IT organizations and storage administrators from delivering consistent storage performance. Storage performance QoS provides IT organizations the ability to scale storage performance consistently without linearly scaling cost and complexity, which will be crucial when moving into a service-centric cloud delivery model.”<sup>1</sup>

IDC

The increasing demands and velocity of change enacted on enterprise IT organizations and service providers today is unrelenting. Users are looking to deploy more applications faster, and for the resources that support those applications to be more agile and adaptive to changing demands. To make things even more challenging for IT, the world’s largest public cloud providers have set the benchmark for what it means to deliver Infrastructure as a Service (IaaS) at scale. So, IT organizations are looking for solutions. They are seeking infrastructure solutions capable of delivering compute, networking, and storage predictably and on demand. Solutions that allow them to dramatically raise operational efficiencies, innovate more quickly, and enable them to respond to application and business challenges faster than ever before.

At the heart of delivering infrastructure on demand, and as a service, is the concept of multi-tenancy, in which multiple applications and/or

customers reside within the same storage infrastructure. While at first glance the opportunity to run a broad array of applications within a single system may sound appealing, the reality for today’s IT managers is very different. When a large number of performance-sensitive applications are consolidated onto a single platform (traditional or flash), “noisy neighbor” applications show up and cause resource contention, unpredictable application performance, and unhappy customers.

So it begs two simple questions:

1. How do you transition from a siloed IT environment to a single infrastructure capable of handling a broad set of applications without sacrificing performance, while at the same time being more agile, automated, and predictable?
2. What storage capabilities are required to do so?

In this guide we look to answer these questions and explore why storage systems with native quality of service (QoS) capabilities have become the key transitional element for companies making the shift to a Next Generation Data Center that is home to applications and workloads that thrive and hum, not lag and freeze.

Eradicating the negative impacts of application performance issues in the data center is no pipe dream. The solution already exists, and it starts with a predictable storage foundation of guaranteed QoS.

1. Storage Array Quality of Service: Provisioning and Guaranteeing Storage Performance, IDC <http://www.solidfire.com/resources/idc-storage-array-quality-of-service-provisioning-and-guaranteeing-storage-performance>

# Introduction to quality of service

“If your primary storage vendor does not have Storage QoS on its roadmap, now is the time to start demanding it.”<sup>2</sup>

Henry Baltazar  
Forrester Research

QoS is a critical enabling technology for enterprise and service providers that want to deliver consistent primary storage performance to business-critical applications in enterprise infrastructure. The type of applications that require primary storage services typically demand greater levels of performance than what is readily available from traditional storage infrastructures today. However, simply providing raw performance is often not the only objective in these use cases. For a broad range of business-critical applications, consistent and predictable performance are the more important metrics. Unfortunately, neither is easily achievable within traditional storage arrays.

There is a large imbalance today between the performance and capacity resources within traditional storage systems. Capacity is plentiful and low cost; conversely, input/output per second (IOPS) are scarce and very expensive. From a provisioning perspective, performance and capacity are rigidly bound together, which only makes matters worse. This bind forces administrators to unnecessarily add storage capacity to increase the amount of IOPS available to a particular application. What results is a wasteful allocation of resources in an effort to overcome the limitations of existing storage architectures.

For service providers and enterprise IT, the promise of delivering storage resources predictably to a broad set of applications without worry has been nothing more than a pipe dream.

## The history

QoS features exist in everything from network devices, to hypervisors, to storage. When multiple workloads share a limited resource, QoS helps provide control over how that resource is shared and prevents the noisiest neighbor (application) from disrupting the performance of all the other applications on the same system.

In networking, QoS is an important part of allowing realtime protocols such as VoIP to share links with other less latency-sensitive traffic. Hypervisors provide both hard and soft QoS by controlling access to many resources including CPU, memory, and network. QoS in storage is less common. If you seek out QoS within the storage ecosystem you will find that most approaches to storage QoS are “soft” – that is, based on simple prioritization of volumes rather than hard guarantees around performance.

Soft QoS features like rate limiting, prioritization, and tiering, are effective only as long as the scope of the problem remains small. When storage is deployed at scale these soft techniques quickly fail. In fact, these features are all “bolt-on” technologies that attempt to overcome limitations in storage architectures that were never designed to deliver QoS in the first place.

<sup>2</sup> Storage QoS is a must-have feature for enterprises and the cloud, [http://blogs.forrester.com/henry\\_baltazar/13-03-13-storage\\_qos\\_is\\_a\\_must\\_have\\_feature\\_for\\_enterprises\\_and\\_the\\_cloud](http://blogs.forrester.com/henry_baltazar/13-03-13-storage_qos_is_a_must_have_feature_for_enterprises_and_the_cloud)

# Where does “QoS” come from?

“Quality of Service,” or “QoS”, originated in the mid-1990s and referred to the overall performance quality experienced by end-users of a telecommunications network.

The term entered the storage realm about five years ago when SolidFire introduced a unique storage architecture specifically designed with the ability to control performance independent of capacity and deliver that performance predictably to thousands of applications within a single storage infrastructure. We now see QoS-like features popping up in the offerings of many storage vendors.

Server virtualization changed the way the world used computing, solving existing problems around inefficiencies, over-provisioning, and cost.<sup>3</sup> But these advances were largely confined to compute and memory, and bypassed the seemingly unaddressable problems associated with storage: unreliable performance and expensive resources.

As a result, we only did half the job. Storage QoS helps with the other half.

Hard QoS controls are defined by rigid terms such as IOPS and MB/s that are strictly enforced and produce predictable results regardless of system function or application activity.

Within the SolidFire platform, each volume is configured with minimum, maximum, and burst IOPS values that are strictly enforced within the system. The minimum IOPS provides a guarantee for performance, independent of what other applications on the system are doing. The maximum and burst values control the allocation of performance and deliver consistent performance to workloads. For the enterprise and service provider, SolidFire QoS enables SLAs around exact performance metrics and complete control over the customer’s experience. For infrastructure consumers, hard QoS delivers clear expectations around storage performance and the ability to deploy all tier 1 and tier 2 applications in the cloud with confidence.

3. Storage Array Quality of Service: Provisioning and Guaranteeing Storage Performance, IDC <http://www.solidfire.com/resources/idc-storage-array-quality-of-service-provisioning-and-guaranteeing-storage-performance>

# QoS: A critical component of the next generation data center

“Storage system capacity is no longer a top concern among the IT professionals I speak with. It’s been replaced with ‘How do I maintain top performance for a given application?’ This is especially true in the virtual environment, where storage I/O is shared. With no guarantee of a specific performance level, mission-critical applications will not be virtualized.”<sup>4</sup>

George Crump  
Storage Switzerland

A quick look across today’s storage landscape shows systems with a broad range of capacity and performance resources. On one end of the spectrum, disk-based systems have a high level of capacity and low level of performance. On the other end, flash architectures deliver a very high level of performance while requiring significantly less capacity (and at much higher cost). When viewed from the application perspective, the reality is that most application performance requirements fall somewhere in the middle of these two storage extremes.

In order to meet varying application performance requirements, the storage industry has responded by implementing caching or tiering schemes in front of traditional disk-based systems. These schemes apply complex algorithms and predictive methodologies that shuffle data to the right media at the right time to boost performance. Costly, complex, and reactive, this approach does little to bring you closer to the predictable performance required by mission-critical applications.

Solving for this disparity requires a more balanced pool of capacity and performance at the system level. From this starting point, a storage system can then deliver performance and capacity scaled independently to serve the unique needs of different applications. This ability to finely allocate capacity and performance resources separately from one another is a fundamental component of next generation data centers.

In these next generation infrastructures raw storage performance is important, but it is the predictable and consistent delivery of that performance which ensures every application has the resources required to run without variance or interruption. In servicing these workloads, IT must consider how well the underlying storage architecture will endure the following conditions:

- Unpredictable I/O patterns
- Noisy neighbor applications
- Constantly changing workload and application performance requirements
- Deduplication, compression, and thin provisioning processes
- Scaling of performance and capacity resources on demand

4. What is Storage QoS? <http://www.networkcomputing.com/storage/what-is-storage-qos/a/d-id/1127906>

# Meet the noisy neighbor

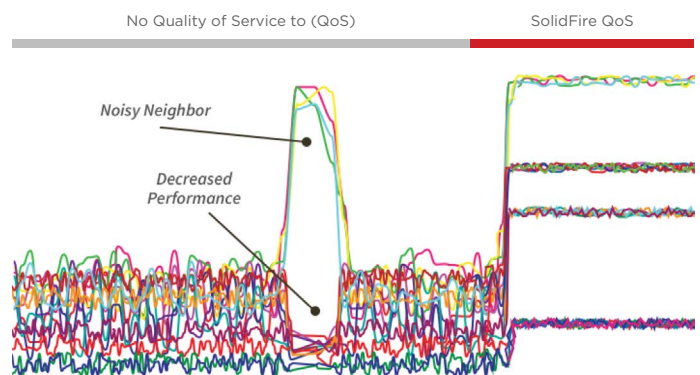
“... customers feel the adverse effects of ‘noisy neighbors.’ This limitation is a major reason why it is so difficult for cloud storage providers [and enterprises] to create consistent multi-tenant environments using traditional storage systems which lack Storage QoS.”<sup>5</sup>

Henry Baltazar  
Forrester Research

Traditionally, when multiple applications share the same storage infrastructure, all performance resources (both IOPS and bandwidth) are freely available to all applications, all the time, across the shared resources (e.g., controller RAID set, set of disk shelves). Without a more precise resource allocation, one application or “noisy neighbor” can easily consume an unfair share of the resources, leaving little available for others. This “first-come, first-served” allocation methodology has a huge negative effect on all of the other applications on the system.

Performance expectations on an application-by-application basis are erratic and unpredictable, a problem that is exacerbated by the poor performance of disk-based architectures when performing random I/O. One misbehaving application can cripple the entire system.

To keep these variances in check, customers must constantly monitor and manage which applications share resources. Often, the solution to alleviating resource contention requires migrating either the “noisy neighbor” or the unhappy customer to a new system.



## Eliminate Noisy Neighbors

Degraded performance from one application spike in a typical multi-tenant infrastructure

When multiple applications share the same storage infrastructure, they also share storage performance (both IOPS and bandwidth). One application—or “noisy neighbor”—can easily consume an unfair share of the resources. Leaving mere scraps for others. SolidFire’s QoS settings eliminate resource contention and variable application performance caused by Noisy Neighbors.

Each volume on the system has been assigned to one of the four levels, providing predictable performance for each application and eliminating the global effect of noisy neighbor activity.

5. Storage QoS is a must-have feature for enterprises and the cloud. [http://blogs.forrester.com/henry\\_baltazar/13-03-13-storage\\_qos\\_is\\_a\\_must\\_have\\_feature\\_for\\_enterprises\\_and\\_the\\_cloud](http://blogs.forrester.com/henry_baltazar/13-03-13-storage_qos_is_a_must_have_feature_for_enterprises_and_the_cloud)

# Not all QoS is created equal

“Without storage QoS, active data will use more storage performance resources, which can starve other storage resources.”<sup>6</sup>

IDC

## If storage performance can be guaranteed, why can't any storage architecture do it?

It's a hard truth to face: legacy storage systems are simply not designed to handle the demands of multiapplication cloud environments. More specifically, the few systems that claim storage QoS – or want to claim it on their roadmap – are really just “bolting it on” as an afterthought. And these “bolted on” methods of achieving QoS such as rate limiting or prioritization have unfortunate side effects.

## Not all QoS is made the same. Not even close.

Most storage vendors are adding QoS features onto existing products, merely solving for one performanceaffecting problem under an isolated condition. This approach falls apart at large scale when conditions multiply, so it's critical that buyers be able to discern True QoS from False.

Let's take a look at some of the current QoS methodologies.

### Tiered storage

How it works – Multiple tiers of different storage media (SSD, 15K rpm HDD, 7.2K rpm HDD) are combined to deliver different tiers of performance and capacity. Application performance is determined by the type of media the application resides on. In an effort to optimize application performance, predictive algorithms are layered over the system which literally try to predict, based on historical performance information, which data is “hot” and kept in SSD vs. data that is “cold” and kept in HDD.

- Performance for every workload varies wildly as algorithms move data between media.
- Variable performance is compounded by uncontrolled noisy neighbors.
- Workloads have no QoS functionality or control over application performance.

Why it doesn't really offer QoS – Tiering is the worst of all the “bolted on” solutions designed for delivering predictable performance. Quite simply, this solution is unable to deliver any level of storage QoS. Tiering actually amplifies “noisy neighbors” because they appear hot and are promoted to higher performing (and scarcer) SSDs, thereby displacing other volumes to lower performing, cold disks. Performance for every workload varies wildly as algorithms move their data between media. No particular application owner knows what to expect of their I/O, as they don't control the tiering algorithm or have any insight to the effect on other workloads. Some tiering solutions try to offer QoS by pinning the data of a particular application into a specific tier, but this approach is essentially dedicated storage (discussed above) at an even higher cost than usual.

### Rate limiting

How it works – Rate limiting attempts to deal with performance requirements by setting a hard limit on an application's rate of I/O or bandwidth. Customers that pay for a higher service will get a higher limit.

- Functionality is designed to protect the storage system, rather than deliver guaranteed QoS.
- Limits are only placed on the maximum performance an application can access.
- There is no concept or capability of delivering performance minimums.
- Applications capped at their max can incur significant latency.

Why it doesn't really offer QoS – Rate limiting can help quiet noisy neighbors, but does so only by “limiting” the amount of performance an application has access to. This one-sided approach does nothing to guarantee the set performance limit can actually be attained. Rate limiting is all about protecting the storage system rather than delivering true QoS to the applications. In addition, firm rate limits set on high performance or bursty applications can inject significant undesired latency.

6. Storage Array Quality of Service: Provisioning and Guaranteeing Storage Performance, IDC <http://www.solidfire.com/resources/idc-storage-array-quality-of-service-provisioning-and-guaranteeing-storage-performance>

## Prioritization

How it works – Prioritization defines applications simply as “more” or “less” important in relation to one another. This is often done in canned and well-defined tiers such as “mission critical,” “moderate,” and “low.”

- Application ranking does not guarantee any application will get the performance it needs.
- Performance is based on arbitrary levels.
- Noisy neighbors can actually get louder if they are prioritized as “mission critical,” monopolizing system performance.

Why it doesn't really offer QoS – While prioritization can indeed help give higher relative performance to some apps and not to others, it doesn't actually tell you what performance to expect from any given tier. It certainly can't guarantee performance, particularly if the problematic “noisy neighbor” is located at the top priority level.

There is no ability to guarantee that any one application will get the performance it needs. What's more, there is no functionality for one application owner to understand what their priority designation means in relation to the other priorities on the same system. It means nothing to tell an owner their application is prioritized as “moderate” unless they know how moderate compares to the other categorizations. Moderate is also meaningless without knowing what system resources are dedicated to a particular tier. Finally, priority-based QoS can actually make a noisy neighbor even noisier if that workload has a higher priority, because it's allowed more resources to turn up the volume.

## Hypervisor-based QoS

How it works – In almost every case, hypervisors are more concerned with “noisy neighbors” than with guaranteeing performance for individual VMs. The hypervisor can use its visibility into the latency and response time for individual virtual disks to set thresholds for when the system will suppress I/O to those VMs that exceed them.

Why it doesn't really offer QoS – The hypervisor, in reality, has very little control or visibility of the underlying storage system resources. Implementing a storage QoS mechanism like storage reservations at the hypervisor layer, without similar enforcement capability at the storage system level, does little to address the core challenges imposed by multi-workload environments. With VMware and others efforting to improve controls at the hypervisor layer, now is the time to demand more from your storage vendors to deliver on their side of this equation.

There is no way for a hypervisor to truly guarantee a minimum IOPS level. In this scenario the hypervisor will always be at the mercy of the storage device.

## Some of the key issues to consider with a hypervisor-centric approach in front of traditional storage include:

### Lack of IOPS control.

While the hypervisor can throttle IOPS, it has no control over maintaining the total I/O pool available. With no governance from the underlying storage system there is no way for a hypervisor to truly guarantee a minimum IOPS level. In this scenario the hypervisor will always be at the mercy of the storage device.

### Performance degradation.

Without visibility into back-end storage resource utilization, there is no way for the hypervisor to know what resources remain available to it on a persistent basis. As storage system utilization increases, performance degradation becomes a real concern. With a larger pool of virtualized applications contending for the same pool of resources, the lack of any sort of storage system layer isolation effectively creates an IOPS free-for-all. The resulting performance variability is a non-starter for infrastructures hosting multiple performance-sensitive applications and workloads.

### Forced over-provisioning.

Absent the ability to granularly carve up storage system performance and provision it out to each virtual machine, the only way to ensure a large enough IOPS pool for these VMs is to extensively over-provision your storage. Unfortunately, there is no better way to blow the economics of your shared storage environment than by being forced to deploy 3x as many systems at one-third the utilization rate.

### Lacking coordination.

While throttling I/O usage to VMs is a basic form of storage QoS, this solution is more of an indictment of the deficiencies of existing storage systems than an ideal solution to the problems faced by large-scale, performance-driven infrastructures. True QoS is delivered through end-to-end coordination and orchestration between the host and the underlying storage system to ensure each virtual machine has the resources it needs to properly support the application.



## Caching

How it works – Caching is the easiest way to reduce contention for a spinning disk. The hottest data is kept in large DRAM or flash-based caches, which can offload a significant amount of I/O from the disks. Indeed, this is why large DRAM caches are standard on every modern disk-based storage system.

Why it doesn't really offer QoS - While caching can certainly increase the overall throughput of the spinning disk system, it causes highly variable latency. Data in DRAM or flash cache can be served in under 1 ms, while cache misses served from disk will take 10-100 ms. That's three orders of magnitude for an individual I/O.

The overall performance of an individual application is going to be strongly influenced by how cachefriendly it is, how large the cache is, and how many other applications are sharing it. In a dynamic cloud environment, that last criteria is changing constantly. All told, it is impossible to predict, much less guarantee, the performance of any individual application in a system based on caching.

## Wide striping

How it works – Once data is placed on a disk, it is seldom moved (except possibly in tiering systems where data is moved to a new tier). Even when a drive fails, all its data is simply restored onto a spare. When new drive shelves are added they are typically used for new data only, not to rebalance the load from existing volumes. Wide striping is one attempt to deal with this imbalance, by simply spreading a single volume across many disks.

Why it doesn't really offer QoS - While this approach can help balance I/O load across the system, many more applications are now sharing each individual disk. A backlog at any disk can cause a performance issue, and a single noisy neighbor can ruin the party for everyone.

The result of this static data placement is uneven load distribution between storage pools, RAID sets, and individual disks. When the storage pools have different capacity or different types of drives (e.g. SATA, SAS, or SSD) the difference can be even more acute. Some drives and RAID sets will get maxed out while others are relatively idle. Managing data placement to effectively balance I/O load as well as capacity distribution is left to the storage administrator, often working with Excel spreadsheets to try and figure out the best location for any particular volume. If the system can't even balance the I/O load it has, how can it guarantee QoS to an individual application as that load changes over time?

# Guaranteed QoS is not a feature — It's an architecture

“Quality of service should not be regarded as a feature that can simply be added to a storage product. QoS functionality that is bolted on after the fact tends to leave conditions in which performance is unpredictable and remains a non-starter for business-critical applications. Complete storage QoS requires [consideration and implementation] at the very core of storage product design.”<sup>7</sup>

Simon Robinson  
451 Research

QoS is a system design choice that must be considered from the very beginning. True QoS delivers predictable performance natively, without having to optimize or organize data layouts to achieve it. Rate limiting, prioritization schemes, and tiering algorithms are all afterthoughts which attempt to overcome limitations in storage systems that were never designed to deliver predictable performance in the first place.

Being able to guarantee performance in all situations – including failure scenarios, system overload, variable workloads, and elastic demand – requires an architecture built from the ground up specifically to guarantee QoS. Trying to bolt QoS onto an architecture that was never designed to deliver performance guarantees is like strapping a jet engine to a VW Beetle. The wheels will come off just when you get up to speed.

The right storage architecture can overcome every predictability challenge by adhering to six core architectural requirements. Together, these six requirements enable true storage QoS and establish the benchmark for guaranteeing performance to every workload.

## **All-SSD architecture**

- Enables the delivery consistent latency for every I/O

## **True scale-out architecture**

- Linear, predictable performance gains as system scales

## **RAID-less data protection**

- Predictable performance in any failure condition

## **Balanced load distribution**

- Eliminate hot spots that create unpredictable I/O latency

## **Fine-grain QoS control**

- Completely eliminate noisy neighbors, and guarantee volume performance

## **Performance virtualization**

- Control performance independent of capacity and on demand

Trying to bolt QoS onto an architecture that was never designed to deliver performance guarantees is like strapping a jet engine to a VW Beetle. The wheels will come off just when you get up to speed.

7. <http://www.solidfire.com/press-releases/solidfire-unveils-benchmark-to-guaranteequality-of-service-in-the-cloud-challenges-storage-industry-to-deliver/>

# The 6 requirements for true QoS

“Traditional storage infrastructures have evolved to better meet the demands of enterprise workloads by leveraging new technologies as they become available. But this is not the same as being purpose-built for the task.”<sup>8</sup>

Aviv Kaufmann  
ESG

Adding QoS features to an existing storage platform may solve one performance bottleneck for individual performance conditions, but this approach fails to solve the exponentially larger challenges that occur at cloud scale. A true solution requires a purpose-built storage architecture that solves performance problems comprehensively, not individually.

In this chapter, we'll dive into greater detail around each of the six required components and capabilities of an IT infrastructure that can enable true QoS.

## Requirement #1: An all-SSD architecture

### What it enables - Delivery of consistent latency for every I/O

Anyone deploying either a large public or private cloud infrastructure is faced with the same issue: how to deal with inconsistent and unpredictable application performance among apps running simultaneously.

The first requirement for achieving this level of performance is moving from spinning media to an all-SSD, or all-flash, architecture. Only an all-SSD architecture allows you to deliver consistent latency for every I/O.

At first, this idea might seem like overkill. If you don't actually need the performance of SSD storage, why can't you guarantee performance using spinning disk? Or even a hybrid disk and SSD approach?

Fundamentally, it comes down to simple physics. A spinning disk can only serve a single I/O at a time, and any seek between I/Os adds significant latency. In cloud environments where multiple applications or virtual machines share disks, the unpredictable queue of I/O to the single head can easily result in orders of magnitude variance in latency, from 5 ms with no contention to 50 ms or more on a busy disk.

An all-flash architecture is just the starting point for guaranteed QoS, however. Even a fast flash storage system can have noisy neighbors, degraded performance from failures, or unbalanced performance.

## Requirement #2: A true scale-out architecture

### What it enables - Linear, predictable performance gains as system scales

Traditional storage architectures follow a scale-up model, where a controller (or pair of controllers) are attached to a set of disk shelves. More capacity can be added by simply adding shelves, but controller resources can only be upgraded by moving to the next “larger” controller (often with a data migration). Once you've maxed out the biggest controller, the only option is to deploy more storage systems, increasing the management burden and operational costs.

This scale-up model poses significant challenges to guaranteeing consistent performance to individual applications. As more disk shelves and applications are added to the system, contention for controller resources increases, causing decreased performance as the system scales. While adding disk spindles is typically seen as increasing system performance, many storage architectures only put new volumes on the added disks, or require manual migration. Mixing disks with varying capacities and performance characteristics (such as SATA and SSD) makes it even more difficult to predict how much performance will be gained, particularly when the controller itself can quickly become the bottleneck.

### Scaling out is the only way to go

By comparison, a true-scale out architecture adds controller resources and storage capacity together. Each time capacity is increased and more applications are added, a consistent amount of performance is added as well. A scale-out architecture ensures the added performance is available for any volume in the system, not just new data. This solution is critical for both the administrator's planning ability as well as for the storage system itself. If the storage system itself can't predict how much performance it has now or will have in the future, it can't possibly offer any kind of guaranteed QoS.

8. Quantifying the Economic Value of a SolidFire Deployment, <http://www.solidfire.com/resources/esg-lab-report-quantifying-the-economic-value-of-a-solidfire-deployment> SolidFire Definitive Guide 14

### **Requirement #3: RAID-less data protection**

#### **What it enables - Predictable performance in any failure condition**

The invention of RAID 30+ years ago was a major advance in data protection, allowing “inexpensive” disks to store redundant copies of data, rebuilding onto a new disk when a failure occurred. RAID has advanced over the years with multiple approaches and parity schemes to try and maintain relevance as disk capacities have increased dramatically. Some form of RAID is used on virtually all enterprise storage systems today. However, the problems with traditional RAID can no longer be glossed over, particularly when you want a storage architecture that can guarantee performance even when failures occur.

#### **The problem with RAID**

When it comes to QoS, RAID causes a significant performance penalty when a disk fails — often 50% or more. This penalty occurs because a failure causes a two- to five-times increase in I/O load to the remaining disks. In a simple RAID10 setup, a mirrored disk now has to serve double the I/O load, plus the additional load of a full disk read to rebuild into a spare. The impact is even greater for parity-based schemes like RAID5 and RAID6, where a read that would have hit a single disk now has to hit every disk in the RAID set to rebuild the original data (in addition to the load from reading every disk to rebuild into a spare).

The performance impact from RAID rebuilds becomes compounded with long rebuild times incurred by multiterabyte drives. Since traditional RAID rebuilds entirely into a new spare drive, there is a massive bottleneck of the write speed of that single drive combined with the read bottleneck of the few other drives in the RAID set. Rebuild times of 24 hours or more are now common, and the performance impact is felt the entire time.

How can you possibly meet a performance SLA when a single disk failure can lead to hours or days of degraded performance? In a cloud environment, telling the customer “the RAID array is rebuilding from a failure” is of little comfort. The only option available is to dramatically under-provision the performance of the system and hope the impact of RAID rebuilds goes unnoticed.

### **Requirement #4: Balanced load distribution**

#### **What it enables - Eliminates hot spots that create unpredictable I/O latency**

Most block storage architectures use very basic algorithms to lay out provisioned space. Data is striped across a set of disks in a RAID set, or possibly across multiple RAID sets in a storage pool. For systems that support thin provisioning, the placement may be done via smaller chunks or extents rather than on the entire volume at once. Typically, however, at least several hundred megabytes of data will be striped together.

Once data is placed on a disk, it is seldom moved (except possibly in tiering systems to move to a new tier). Even when a drive fails, all its data is simply restored onto a spare. When new drive shelves are added they are typically used for new data only, not to rebalance the load from existing volumes.

Wide striping is one attempt to deal with this imbalance, by simply spreading a single volume across many disks. But when combined with spinning disk, wide striping increases the number of applications affected when a hotspot or failure does occur.

#### **Unbalanced loads cause unbalanced performance**

The result of this static data placement is uneven load distribution between storage pools, RAID sets, and individual disks. When the storage pools have different capacity or different types of drives (e.g. SATA, SAS, or SSD) the difference can be even more acute. Some drives and RAID sets will get maxed out while others are relatively idle. Managing data placement to effectively balance I/O load as well as capacity distribution is left to the storage administrator, often working with Microsoft Excel spreadsheets to try and figure out the best location for any particular volume.

Not only does this manual management model not scale to cloud environments, it just isn't viable when storage administrators have little or no visibility to the underlying application, or when application owners cannot see the underlying infrastructure.

The unbalanced distribution of load also makes it impossible for the storage system itself to make any guarantees about performance. If the system can't even balance the I/O load it has, how can it guarantee QoS to an individual application as that load changes over time?

### **Requirement #5: Fine-grain QoS control**

#### **What it enables - Complete elimination of noisy neighbors and guaranteed volume performance**

Another key requirement for guaranteeing QoS is a finegrain control model that describes performance in all situations. Contrast fine-grain control against today's rudimentary approaches to QoS, such as rate limiting and prioritization. These features merely provide a limited amount of control and don't enable specific performance in all situations.

#### **The trouble with having no control**

For example, basic rate limiting, which sets a cap on the IOPS or bandwidth an application consumes, doesn't take into account the fact that most storage workloads are prone to performance bursts. Database checkpoints, table scans, page cache flushes, file copies, and other operations tend to occur suddenly, requiring a sharp increase in the amount of performance needed from the system. Setting a hard cap simply means that when an application actually does need to do I/O, it is quickly throttled. Latency then spikes, and the storage seems painfully slow, even though the application isn't doing that much I/O overall.

Prioritization assigns labels to each workload, yet similarly suffers with bursty applications. While high priority workloads may be able to easily burst by stealing resources from lower priority ones, moderate or low priority workloads may not be able to burst at all. Worse, these lower priority workloads are constantly being impacted by the bursting of high priority workloads.

Failure and over-provisioned situations also present challenges for coarse-grain QoS. Rate limiting doesn't provide any guarantees if the system can't even deliver at the configured limit when it is overtaxed or suffering from performance-impacting failures. While prioritization can minimize the impact of failures for some applications, it still can't tell you ahead of time how much impact there will be, and the applications in the lower tiers will likely see horrendous performance.

### **Requirement #6: Performance virtualization**

#### **What it enables - The ability to separate provisioning for capacity and provisioning for performance — on demand**

All modern storage systems virtualize the underlying raw capacity of their disks, creating an opaque pool of space from which individual volumes are carved. However, the performance of those individual volumes is a secondorder effect, determined by a number of variables such as the number of disks the volume is spread across, the speed of those disks, the RAID-level used, how many other applications share the same disks, and the controller resources available to service I/O.

#### **Traditional capacity virtualization does not suffice**

Historically this approach has prevented storage systems from delivering any specific level of performance. "More" or "less" performance could be obtained by placing a volume on faster or slower disks or by relocating adjacent applications that may be causing impact. However, this solution is a manual and error-prone process. In a cloud environment, where both the scale and the dynamic nature prevent manual management of individual volumes, this approach just isn't possible. Worst of all, significant raw capacity is often wasted as sets of disks get maxed out from a performance standpoint well before all their capacity is used.

# SolidFire's Guaranteed QoS

The burden of acquiring and managing multiple disparate storage infrastructures is about to become unbearable (if it has not already). But when you consolidate all of these applications on to a single storage system you run the risk of having too much performance variability to ensure any particular workload gets the performance it needs. So what do you do?

SolidFire's QoS technology is focused on enabling large enterprises and service providers to assign and guarantee fine-grain levels of performance (IOPS and bandwidth) to thousands of volumes residing within a single storage platform

This approach proactively provides applications with the performance they require from day one throughout the life of their deployment. With guaranteed QoS from SolidFire, applications no longer contend for performance, and administrators no longer have to hassle with complex tiering systems or prioritization schemes.

SolidFire's fine-grain QoS controls stem from our patented performance virtualization technology. This unique technology enables service providers and enterprises alike with two key operational functions:

1. The ability to control performance and capacity independently from one another
2. The ability to set fine-grain guaranteed QoS levels on a per-volume basis

Within a SolidFire storage array, performance and capacity are presented as independent unified pools that are entirely separate from one another. Each storage volume within the system can be allocated an exact amount of capacity and performance, both

of which can be changed on the fly without migrating data or impacting performance.



## All-SSD Architecture

Enables the delivery of consistent and see for every I/O



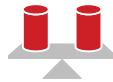
## True Scale-Out Architecture

Linear, predictable performance gains as system scales



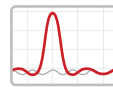
## Raid-less Data Protection

Predictable performance in any failure condition



## Balanced Load Distribution

Eliminate hot spots that create unpredictable I/O latency



## Fine-Grain to QoS Control

Completely eliminate noisy neighbors and guaranteed volume performance

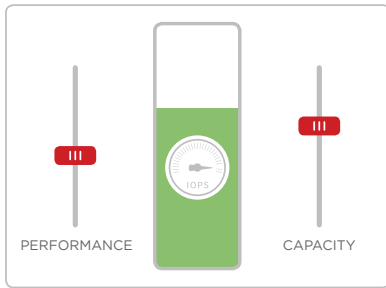


## Performance Virtualization

Control performance independent of capacity and on demand

Only with SolidFire, manage performance predictably and independent of capacity.

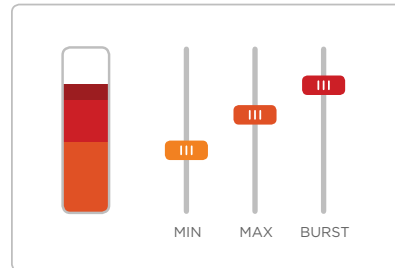
SolidFire's patent-pending performance virtualization technology allows for the fine-grain allocation of performance without incurring the capacity sprawl and low utilization rates common with traditional disk-based systems.






*Control performance and capacity independent of each other.*

SolidFire's performance virtualization technology allows for the fine-grain allocation of performance without incurring the capacity sprawl and low utilization rates common with traditional disk-based systems.

Allocate, manage, and guarantee storage performance



- 
 → Min IOPS  
 IOPS that are always available to the volume. Ensures guaranteed performance regardless of system condition or application activity.
- 
 → Burst IOPS  
 IOPS that a volume will be allowed to process during a spike in demand. Particularly effective for uneven and latency sensitive workloads.
- 
 → Max IOPS  
 IOPS that a volume can process over a sustained period of time.

Volumes provisioned within a SolidFire system are assigned three performance values: max IOPS, burst IOPS, and min IOPS. Each value can be monitored, tracked for chargeback, and changed on the fly without impacting volume or system performance.

- Max IOPS is the maximum number of sustained IOPS a volume will be allowed to process over an extended period of time. Applications will not be permitted to consistently exceed this level and affect other applications.
- Burst IOPS is the maximum number of IOPS a volume will be allowed to process over a short period of time. When a volume uses less than its max IOPS it will accumulate credits, which can be used to burst to a volume’s burst IOPS limit for a short period of time. Burst IOPS is particularly effective for virtual machine reboots, migrations, large file transfers, and other heavy loads that need to be completed within a short period of time. This functionality is only allowed when system performance resources are available, preventing any impact on other applications.
- Min IOPS is the minimum number of IOPS that an administrator grants to a volume. This IOPS level is what is effectively “guaranteed” and is the focus of most conservative service-level agreement (SLA) provisions. Min IOPS values come into play only if the system becomes bound by I/O capacity, at which point the system will scale all volumes back from their max IOPS level proportionally toward their min IOPS values. This ensures fair resource allocation when the system is heavily loaded and also offers a prioritization mechanism to give more important volumes priority at times of heavy load, while others are scaled back more dramatically.

In all cases the min IOPS setting ensures a predictable level of performance rather than the random performance degradation typically seen in performance-constrained situations. Note that the “guarantee” is limited by the I/O capacity of the system; if the total min IOPS of active volumes exceeds the I/O capacity of the system (i.e., oversubscription), performance will continue to scale down proportionally.

As QoS becomes a must-have component of a storage infrastructure, the differences between QoS features and a purpose-built QoS architecture become evident. SolidFire’s all-flash storage system is purpose-built to enable IT organizations to allocate, manage, and guarantee storage performance — making it faster and easier to respond to changing demands of applications and the business than ever before.

**Create New Volume**

Volume Name :

Account :

Total Size :

Enable 512 Byte Emulation

**Quality of Service Settings**

IO Size	Min	Max	Burst
4 KB	1000 IOPS	2000 IOPS	8000 IOPS
8 KB	625 IOPS	1250 IOPS	5000 IOPS
16 KB	370 IOPS	741 IOPS	2963 IOPS
256 KB	26 IOPS	51 IOPS	205 IOPS
<b>Effective Max Bandwidth</b>		13.98 MB / sec	55.92 MB / sec



# QoS for service providers: New services and opportunities

Why should service providers care about true storage QoS? The unique design of public multitenant infrastructures leads to more inherent challenges than those experienced by a traditional enterprise. While enterprise workloads are typically more standard in nature, the workloads that run in hosting environments are largely unknown and unpredictable.

Most service providers have no real knowledge of what kinds of applications or workloads their enterprise customers are running. Without QoS, applications can run rampant and quickly become noisy neighbors. For the service provider, performance variance caused by a noisy neighbor can have huge consequences if those neighbouring applications are that of another company or customer.

Because hosting business-critical applications in the cloud represents a large revenue opportunity for service providers, the ability to deliver predictable hosting services free of noisy neighbors is a critical capability. 451 Research states that only 32% of enterprise applications are running in a hosted infrastructure, and until storage performance is predictable and guaranteed, service providers won't be able to programmatically deliver services that attract the other 68% of those enterprise workloads.

Is there a solution? Yes, and the answer is storage QoS architected from the ground up with guaranteed performance in mind.

Another benefit of storage QoS and the ability to guarantee a minimum level of performance to every application is the ability to offer firm SLAs on storage performance. The storage system's ability to guarantee a minimum level of performance makes writing SLAs a snap. Regardless of system condition or an application's activity, performance is guaranteed and has become a surefire way to attract new enterprise hosting revenue.

Service providers should be looking at QoS with the long-term goal of writing firm SLAs against storage performance. Without it, they will remain unable to efficiently meet the rising performance requirements of enterprise customers looking to host their businesscritical applications.

## Setting performance SLAs

Although IOPS settings and enforcement are the basis of ideal QoS capability, administrators should consider additional operational factors when setting performance expectations and related SLAs for internal (enterprise) and external (service provider) customers. When developing performance-based SLAs, consider the four key areas that follow.

## System provisioning

At the heart of any SLA strategy is a stance on storage volume provisioning. Is your strategy to be aggressive with provisioning (i.e., heavily oversubscribed)? If so, this would dictate a more conservative SLA strategy. However, if provisioning is done in a very conservative manner, there is headroom to take a more aggressive stance with SLAs.

## Understanding total system load

When crafting an SLA, it is important to consider factors beyond the purely quantitative IOPS metrics. An IOPScentric approach fails to fully capture the impact of varying block sizes on overall performance. Accounting for the performance-related impact of varying block sizes requires a more comprehensive approach, one in which the concept of system load comes into play. System load is a function of IOPS and average block size. Incorporating these two variables into a more holistic metric produces a more accurate indication of the actual load being placed on the system.

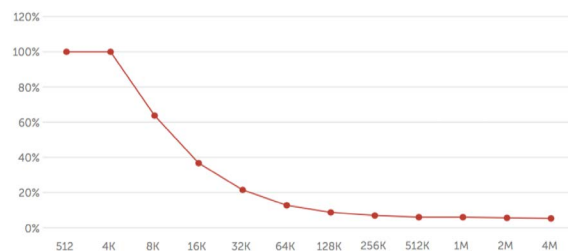


Exhibit A: The SolidFire Load Curve

## Load balancing

In cloud block storage, the most frequently encountered workload involves large amounts of small random I/O spread out across numerous application volumes. However, it is also critical to understand and account for the potential performance impact of other workload profiles (for example, in an instance where there is high concentration of I/O into a small number of volumes). Well-written SLAs create awareness and establish appropriate expectations around these outliers.

## Impact of failure conditions

When writing performance-based SLAs, contemplate the impact of component-level failure (e.g., disk drive) on both capacity and system-level performance. One way to account for any potential performance degradation under failure conditions is to create a performancelevel guarantee within the SLA, which would commit to a specified level of performance for a percentage of time (e.g., 95%). The buffer outside of this performancelevel (e.g., 5%) leaves appropriate headroom to absorb performance degradation under certain failure conditions.

# The impact of QoS in the enterprise

Enterprises today are tasked with figuring out how to build a flexible, scalable platform that can support multiple workloads while improving operational efficiency. Up until now, storage administrators have spent the bulk of their time tuning, tweaking, planning, and troubleshooting storage performance.

They continue to face several pain points:

- Identifying and protecting applications that have different I/O patterns
- Managing separate, siloed storage appliances, each corresponding to a separate workload
- Overcoming the difficulty of sizing storage for both initial workload placement and growth over time
- Eliminating inefficiencies and waste in capacity, performance, and operational management

It doesn't have to be this way. The technology and know-how exists to have predictable, flexible, and easily managed storage as part of an overall virtualized platform designed to provide the performance and availability that today's end user workloads demand.

It all comes down to this: enterprises need more flexibility, and a solution that allows them to provision capacity and performance separately and uniquely for every application, every time. QoS gives them this ability.

One of the most effective ways enterprise customers are taking advantage of QoS is by consolidating multiple workloads, typically ones that have been previously isolated from one another in separate storage silos. By allowing for many applications to be deployed onto a single platform with guaranteed QoS, enterprise IT can now easily address all performance-related challenges within a single storage system.

- By reducing the number of storage platforms and vendors in use, the cost of operations goes down, and the number of tools needed to manage storage is decreased.
- By provisioning capacity and performance separately out of a single pool, less over-provisioning is required in order to meet the needs of the workloads.
- By providing a scalable platform that can grow or shrink based on the collective needs of the business, enterprises can make more efficient use of capital, space and power, and manpower.

Nowhere is this concept of consolidation more powerful or relevant than in the virtualized infrastructures of today's enterprise IT. Being able to provision capacity and performance separately from a storage platform finally unifies the resource management processes in a way that hasn't been available before. It's just like how enterprises have been using cloud management systems to provision CPU and RAM separately for years.

The ultimate effect is to drive more efficiency, more integration, better performance, and improved availability for the workloads while reducing the burdens of management for the operations teams.

The Enterprise Strategy Group (ESG) recently published a lab report in which they compared a SolidFire system with a traditional storage vendor's well-documented reference architecture designed for sizing VMware deployments.<sup>9</sup> For customers looking to consolidate multiple workloads, the report demonstrated a strong TCO and outlined how customers can guarantee performance, deploy far less hardware, and simplify their systems management.

ESG Lab's cost/benefit analysis indicated that by virtualizing and automating performance, SolidFire can eliminate up to 93% of traditional storage-related problems. These problems might include issues inherent in a traditional architecture that are caused by workload imbalance, monopolization of a fixed set of resources, insufficient resources in a pool, requirements to move VMs, inefficient tiering, and controller bottlenecks. The report also highlighted SolidFire's ability to lower operating expenses by up to 67% by automating many of the time-consuming tasks performed by traditional storage administrators, and estimated customers could more rapidly respond to the demands of the business by deploying VMs up to 15 times faster than traditional storage architectures.

9. Quantifying the Economic Value of a SolidFire Deployment, <http://www.solidfire.com/resources/esg-lab-report-quantifying-the-economic-value-of-a-solidfire-deployment>

# Conclusion

Now that you've learned how guaranteed QoS is the transformative element in next generation data centers, it's time to start moving toward it. Use the tips and benchmarks provided in this guide, and get even more guidance by starting a conversation with SolidFire.