White Paper
Enterprise Software Defined Block and Object Storage
- Built for Performance -

Table of Contents

List of Figures

Preface

This document describes the basic architecture of the ETERNUS DSP system and provides the features of the enterprise storage system by explaining the product specifications, benefits, and tradeoffs.

The product lineup and product information stated in this document are current as of January 2020.

■Intended Audience

This document targets the following audience:

- Those who are considering installation or replacement of storage systems
- Those who are proposing installation or replacement of storage systems

■Applicable Model

This document covers the following model.

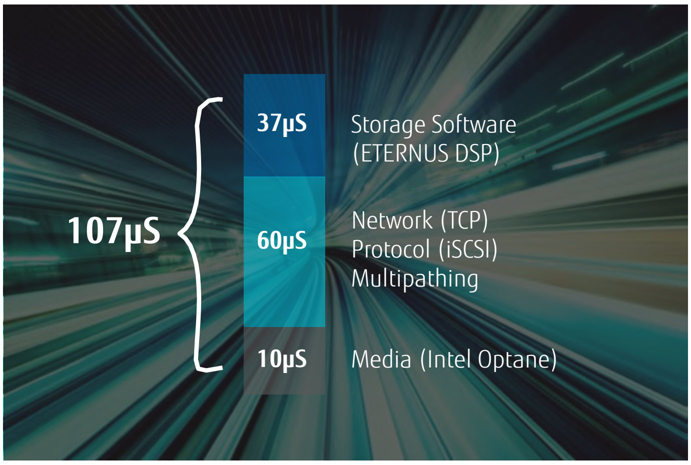
- FUJITSU Storage ETERNUS Data Services Platform Software

■Naming Conventions

The following abbreviations are used in this document.

- ETERNUS DSP	FUJITSU Storage ETERNUS Data Services Platform Software
- Web GUI	Web GUI for ETERNUS DSP
- SSD	Solid State Drive

1. Introduction



Enterprises implementing a software defined data center (SDDC) need more out of their storage, a lot more. Gone are the days of promoting performance as a theoretical set of IOPs on a contrived implementation under idealized conditions. In the SDDC, data storage performance is one attribute among many, to be set programmatically, managed autonomously and changed dynamically.

This paper covers how ETERNUS DSP's architecture enables you to achieve Tier 1 enterprise performance on your toughest workloads, while still enabling cloud-like agility on premise.

ETERNUS DSP's autonomous data management platform was architected from the outset to deliver, not only unprecedented, low latency / high IOP performance for real world SDDC environments, but without compromising on the other critical infrastructure needs, including automation, continuous availability, data management and the ability to continuously adopt new technology as it becomes available.

In order to understand data storage performance, we must first talk briefly about the objectives of the SDDC. Two essential requirements of today's SDDC are developer velocity and operational agility.

- **Developer Velocity.** Supporting the rapid development and deployment of applications requires a robust and programmable platform to enable application development and testing as well as quick deployment to production, scaling, and protection and migration. The SDDC must support an application through its entire lifecycle, from birth to death, and the underlying infrastructure must enable the application lifecycle to transition through its phases based on business needs, not hinder it. The infrastructure should provide any and all services needed by the application such that it instills the confidence to move quickly and potentially to create business advantage, which is all about velocity.
- **Operational Agility.** Supporting operational agility requires infrastructure that is rich in capability, flexible in implementation and autonomous in operation. Key among these data storage capabilities is performance, but performance cannot be considered in isolation from the other factors essential within the overall data center environment. Instead, the infrastructure should enable a full complement of services, including replication, deduplication, encryption, compression, protection, performance and beyond. As important, the infrastructure should enable these services to be changed as the needs of the business demand changes, on-the-fly and without disruption. Furthermore, it must enable the ability to scale, upgrade and adopt new technology, again on-the-fly and without disruption, which yields agility.

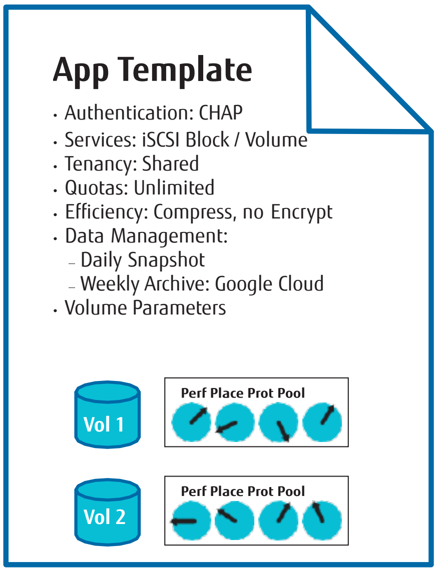
Historic measures of performance often prove inadequate in guiding enterprises to evaluate and select the optimal storage infrastructure for their SDDC deployment. This paper describes a richer model for talking about data storage performance in the SDDC.

2. Applications Drive the Software Defined Data Center

Inherent in the SDDC is the notion that there are many applications, many deployment models (bare metal, virtual machine, container) and many tenants all operating simultaneously at different stages in their lifecycle. So it is essential that the infrastructure selected can treat each individually, since they inevitably evolve at different rates and change frequently as the needs of the business and use base change. Change is the only constant within the SDDC, since new applications and technology often are introduced just as other applications and technology are being retired, which is why ETERNUS DSP was built for change (please see ETERNUS DSP's Built for Constant Change whitepaper for further discussion).

To model storage performance in the SDDC, let's start by breaking down the SDDC into application instances that in aggregate make up the SDDC performance ***demand***. To do this, we must specify the requirements of an application instance in an application template. A template abstracts the specific needs of application instance, enabling developers (the "Dev" in DevOps) to instantiate an application instance on-demand and programmatically without requiring them to understand the underlying mechanisms. In addition to minimizing storage-specific skill requirements from developers, templates can be programmatically (or manually) changed as business needs evolve and the ETERNUS DSP system will autonomously adjust to meet those new needs.

Within ETERNUS DSP, application templates are translated into a set of policies associated with the application instance data. These policies are then converted internally into a Service Level Agreement (SLA). During operation, telemetry information is gathered and analyzed throughout the data center and continuously compared to the SLA to ensure each parameter is met and to make adjustments when they are needed. This creates a closed loop system wherein the ETERNUS DSP platform is constantly monitoring the environment and reporting on SLA achievement. Note that using the application templates, the policies and therefore the associated SLAs can be changed at-will, based on business needs and the system will autonomously adjust to meet the new ones.



App Template

- Authentication: CHAP
- Services: iSCSI Block / Volume
- Tenancy: Shared
- Quotas: Unlimited
- Efficiency: Compress, no Encrypt
- Data Management:
  - Daily Snapshot
  - Weekly Archive: Google Cloud
- Volume Parameters

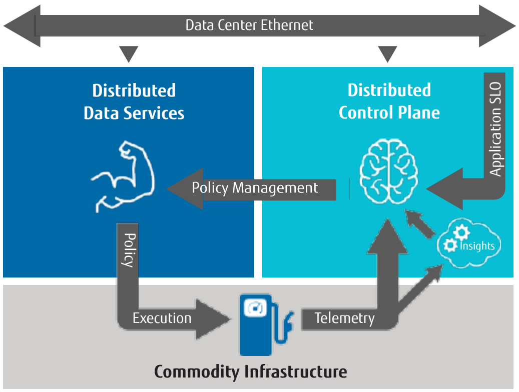
It is important to point out that an application instance is a construct describing all the data volumes, protection, resilience, fault domains, data management, performance, accessibility for an application. Unlike other systems, this is not done on a volume by volume, but more simply and holistically done on a class basis, eliminating much of the individual tuning associated with past infrastructure options.

3. A Model for Storage Performance in the SDDC

In the SDDC, each application instance consumes storage resources, including performance. The SDS must support two elemental performance demands: both instantaneous performance demands and transitional performance demands. This is essential because due to the changing nature of performance over the course of an application's lifecycle.

So let's examine the two elements of performance:

- Instantaneous performance represents the IOPs, latency, and throughput at a given point in time that are required to satisfy the needs of an application instance, inclusive of the required data management attributes such as deduplication, encryption and snapshots for protection.
- Transitional performance represents the IOPs, latency and throughput required during a fundamental change in the system, i.e. a transitional state. Examples of transitional change include a modification of the policies used to manage the application lifecycle or a modification in the infrastructure itself, including recovery from failure, adding capacity or new technologies (e.g., new media, new generations of CPU).



In a multi-tenant, service-oriented architecture, ideally the management of the infrastructure and the management of tenants will have no effect on the instantaneous performance of a single application instance. Said differently, the service level agreement for one application instance should not be impacted by other changes made for other application instances or underlying changes to the infrastructure.

4. Performance is Part of the Service Level Agreement

The application instance SLA is an essential element used to deliver the desired experience to the application owner. It specifies the level of performance, resilience, protection and efficiency via policies that are then autonomously managed by the ETERNUS DSP software. And as mentioned previously, ETERNUS DSP enables the ability to change the policies on-the-fly as the application needs change throughout the application lifecycle.

ETERNUS DSP manages each application instance via the SLA, and performance against individual application instances SLAs can be evaluated as well as the aggregate telemetry and statistics per tenant and per system, respecting the administrative rights of the user.

For each application instance, performance attributes are simplified into performance tiers in the system, often named platinum, gold, silver and customized by organization, which in turn are utilized to place replicas on the appropriate storage nodes and medium. This placement also takes into consideration other SLA attributes including fault domains, data management (snapshot, deduplication, compression, encryption) and resilience (replication count).

It is critical to point out that an application instance is much more than a volume or a LUN. Most applications instances have multiple types of data that require different capacities or types of storage media. For example, a database, in addition to row and columnar data, typically has various indexes and logs. These different data types may benefit from different data management, data protection and performance, since application performance is often dictated by metadata performance.

5. Supply and Demand

We can think about performance using two simple principles, supply and demand. Supply is denoted by the Storage Infrastructure Performance Potential or SIPP. SIPP can be calculated as a function of compute, network and storage hardware as well as software components that make up the infrastructure for an ETERNUS DSP cluster or system. The configuration of a cluster establishes the SIPP and SIPP establishes a supply that is consumed by instantaneous and transitional demands by both application instances and infrastructure management. In historic terms, the SIPP was often quoted as the performance of a system.

The problem with quoting the SIPP is that it does not take into account many real world effects such as workload variability, co-mingled demand, error recovery and environmental bottlenecks. SIPP remains a useful construct in describing potential but less useful in helping a customer meet their actual business needs. It is far better to consider SIPP as a supply from which to meet a demand at a given point in time.

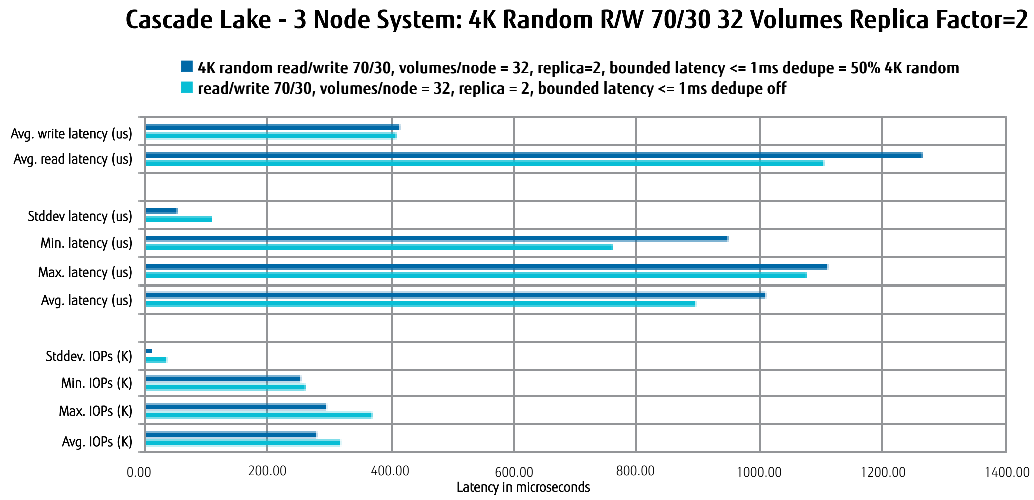
Demand is denoted by the aggregate demand of the application instances, management operations and transitional load.

You can think of supply and demand as a busy highway. During rush hour, demand is high, perhaps exceeding supply. Throw in a lane closure (system maintenance), an accident (media failure), or construction of a new lane (scaling), and the situation worsens due to supply constraints. Storage systems are not unlike a road system, moving data from place to place, as quickly and efficiently as possible.

6. Performance Measures

Historically performance has been viewed through three common metrics:

- **IOPs** – the measure of ***how many*** I/O operations can be done in a specified period of time
- **Throughput** – the measure of ***how much*** data can be moved in a specified period of time
- **Latency** – the measure of ***how long*** it takes to complete a typical I/O operation



Configuration

- Number of nodes: 3
- Memory (GB): 376
- CPU gen: Cascade Lake
- NVDIMM: jedec_agiga_dsm
- Storage: SATA Flash
- Number of clients: 3
- Number of volumes: 32
- Replica count: 2

Figure 6-1 Node ETERNUS DSP Cluster with SATA SSDs Latency & IOPs

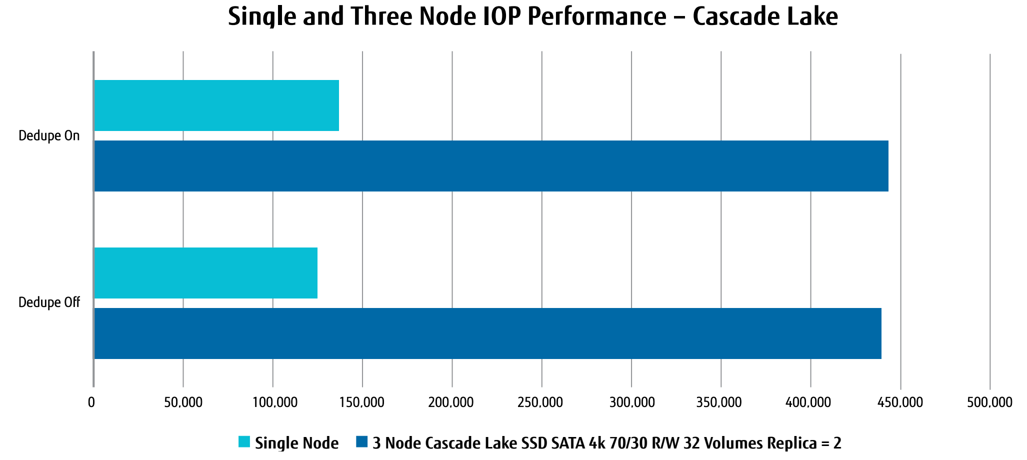
More recently, an overlay metric has emerged, namely predictability. Performance predictability is the percentage of a performance metric delivered within bounds, which has become all the more critical in the SDDC handing a large volume of applications and tenants. Most often this metric is applied to latency, but it can be applied to IOPs and throughput as well, but for the purpose of this paper predictability will be focused on latency as this is typically what impacts application performance.

The industry has created a term called tail latency to converse about latency predictability. Tail latency denotes the "long-tail" of latency distribution where some fraction of the IOPs are taking inordinately long (orders of magnitude difference) to complete relative to the latency distribution of the other IOPs.

For example, a system might complete 10 million IOPs with an average latency of 500 microseconds, but two of the 10M IOPs took 5 seconds to complete. The average latency is largely unaffected by the two IOPs that took 5 seconds, but the application may have failed or initiated some sort of recovery due to these two outliers. In this example, the tail latency varied by 4 orders of magnitude!

7. Calculating Supply

Performance supply is the amount of performance available at a given moment to serve aggregate application instance and infrastructure demands. For the purpose of this paper we will focus on IOPs, latency and predictability. Throughput Is not the focus of this paper as most applications served by the ETERNUS DSP system focus on IOPs and latency.



Configuration

- Number of nodes: 3
- Memory (GB): 376
- CPU gen: Cascade Lake
- NVDIMM: jedec_agiga_dsm
- Storage: SATA Flash
- Number of clients: 3
- Number of volumes: 32
- Replica count: 2

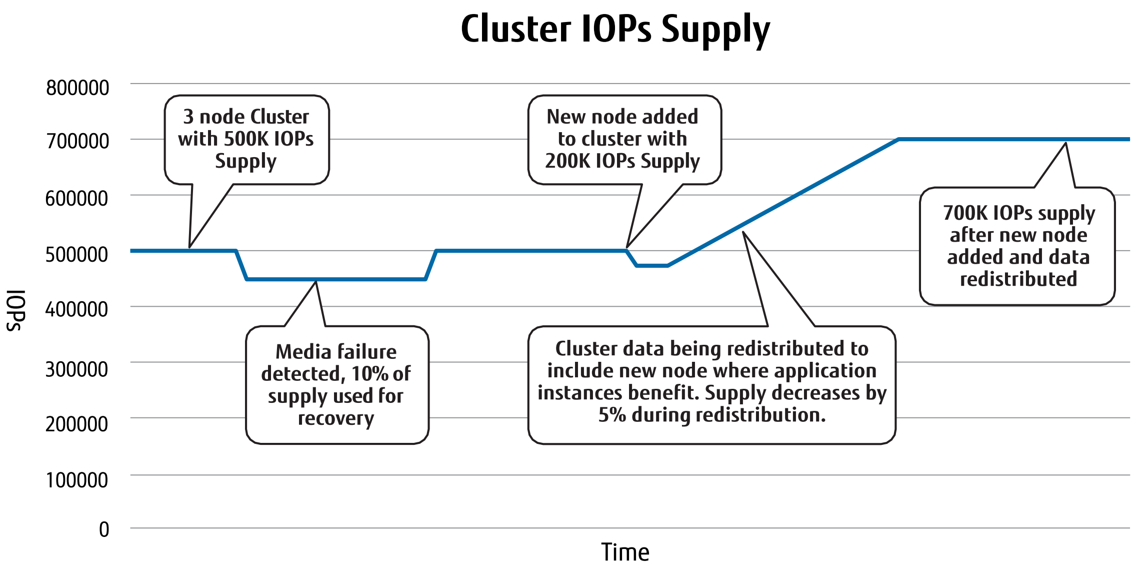
Note that ETERNUS DSP performance is minimally impacted by our adaptive dedupe engine.

Figure 7-1 Node SATA Flash Dedupe On vs. Off

With modern solid-state media (NAND Flash, 3D Xpoint), performance limitations are typically outside of the media, often a combination of processor, network of software limitation. In HDD based systems, the physics of the HDDs were the primary limitation, particularly on IOPs and latency.

ETERNUS DSP, as part of our Hardware Compatibility List (HCL), will calculate the SIPP of each node type supported and the aggregate of each node's SIPP can be used as a first order calculation of the cluster performance potential. ETERNUS DSP recommends that the cluster communication network have 4X the bandwidth of the application network, i.e. a 25gbit front end application network should be supported by the 100gbit back-end cluster network. It is possible to use a single (redundant) network for both application and cluster traffic and the expectation should be that application performance will be limited to approximately ¼ of the cluster traffic. For example, a 100gbit network would support 20gbit of application traffic with 80gbits being used for cluster communication (replication traffic primarily).

A cluster consisting of three nodes, each with a 166K IOPs potential, and enough network performance, the total cluster SIPP would be 500K IOPs. But that is not the end of the story… now we must consider how this supply is impacted by typical operations over time.



If you review the chart above, you will see an example of how supply is impacted during a media failure and scaling the cluster. When media fails, the data contained on the media is retrieved from other replicas in the cluster and redistributed within the cluster in conformance to the polices (performance, fault domains,…) for the application instance. This redistribution creates a transitional demand which reduces the available supply of performance for other needs, namely application instances and other management operations. The ETERNUS DSP software throttles resource consumption during these kinds of operations with this consumption, set by default, to consume no more than 10% of the SIPP. This insures minimal impact on applications instances while still guaranteeing progress of management operations. Once the media failure is remediated the supply returns to normal.

During scaling, i.e. when nodes are added or removed from the cluster, there is also transitional demand that impacts SIPP. In the chart above a new node is being added that will add another 200K IOPs to the SIPP. Once the node is added to the cluster, the policies associated with the applications instances are evaluated and for any application instance where the SLA would benefit from consuming resources associated with the new node, data is migrated to the new node. As this occurs SIPP increases incrementally until the SIPP of the new node is fully realized within the cluster.

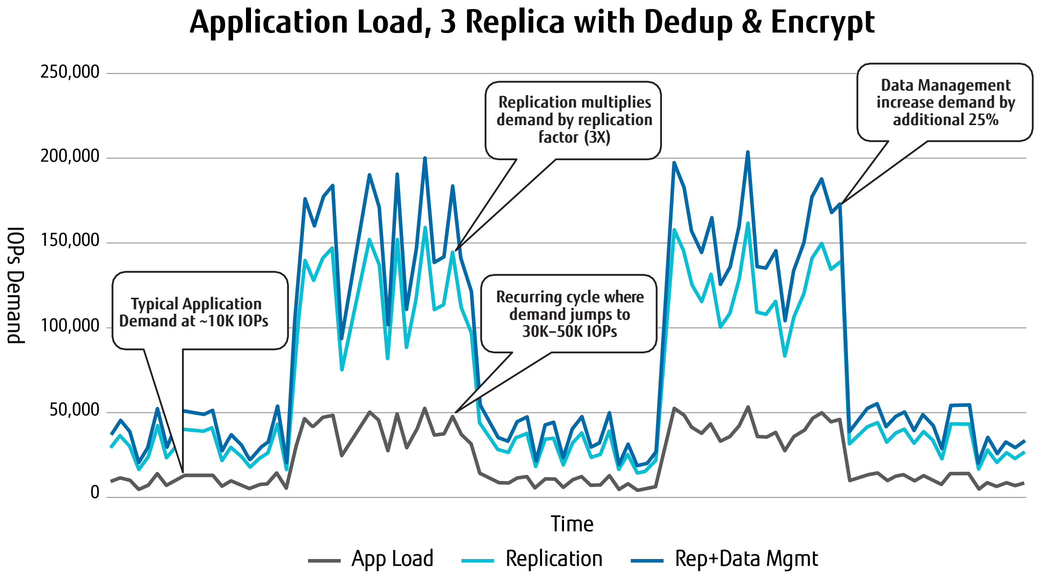
In large environments, management operations are commonplace and, as a result, SIPP will naturally vary over time. This variation should be carefully considered in sizing a cluster for a given aggregate workload – which brings us to the topic of ***demand***.

8. Calculating Demand

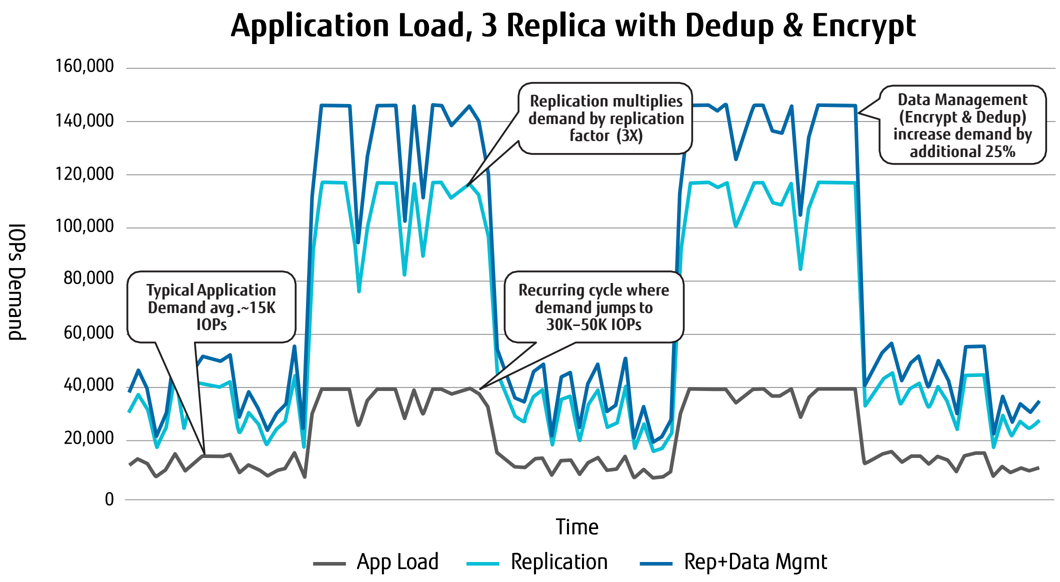
Demand is simply the amount of work required at a point in time. Demand is created by the aggregate of the following:

- Application instance workload including
  - I/O Rate (per volume)
  - Read / Write Ratio
  - Replication factor (per volume)
  - Compression (per volume)
  - Deduplication (per volume)
  - Snapshot schedule (all volumes in app instance)
- Encryption (per cluster)
- Migration (per volume caused by policy change, error recovery, scaling)
- Error Recovery (network, media, node, software)
- Upgrade (rolling upgrade disruption)
- Scaling (related migration)

While it is relatively easy to calculate supply, and the sort of events that cause supply to vary, it is often very difficult to estimate demand beforehand. Difficulty in calculating demand stems from two key variables that are often not well understood: 1) cyclical variability in application instance demand 2) multiplicative effect of application demand resulting from data protection and management.



The chart above illustrates the wide variation changes in application workload can have on overall demand. In the chart, an application has a typical demand of 10K IOPs with a 50% variation. In the time measured, two cycles occur where the demand increases 3X with a commensurate increase in variation. This translates to a front-end cyclical demand of up to 50K IOPs. With a replication factor policy of 3X replicas, this front-end demand can spike up to 150K IOPs. Data management functions such as compression, deduplication and encryption can be modeled as consuming resources that translates into demand that can be approximated as IOPs and in the chart above contribute an additional demand of 25%.



In the chart above, the application instance had a Quality of Service limit set of 40K IOPs (front-end demand). Setting QOS limits ensures an application cannot create excess demand in the system, in this case reducing the maximum IOPs demand from ~200K IOPs to 140K IOPS. Note that this does not decrease the number of IOPs performed by the system, it spreads the IOPs over a greater amount of time, thereby increasing latency for the particular application instance when the QOS threshold is exceeded. Spreading IOPs over greater time under heavy load leads us to the next topic – predictability.

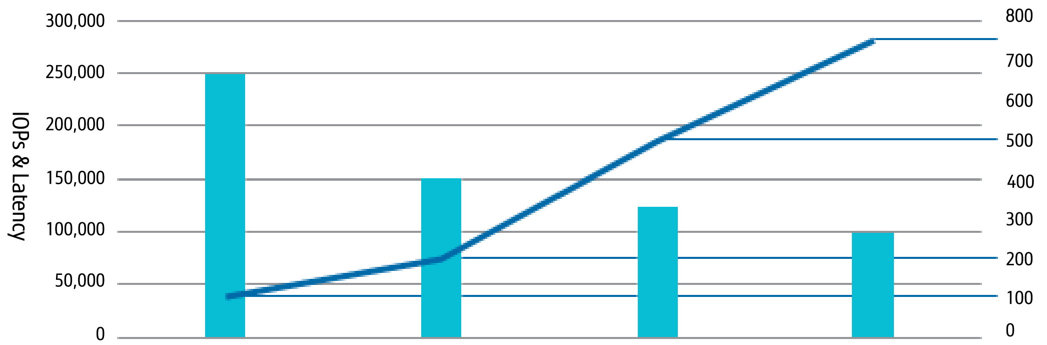
9. Predictability

As mentioned above, performance predictability is the percentage of a performance metric delivered within a bounds or range. Based on customer interaction, this paper focuses on latency predictability.

The question for latency predictability is variation, how much and how often.

Predictability behavior, distilled to its essence, comes down to the fact that when supply is proportionally high relative to demand the cluster will operate with greater predictability. For example, if demand is less than 50% of supply the ETERNUS DSP cluster would deliver 99.99% of IOPs within one order of magnitude of the average latency. If average latency were 500us, then 99.99% of the IOPs would complete between 50us and 5ms. Conversely, if demand gets past 90% of supply, creating resource contention, predictability will suffer such that 99.99% of the IOPs may range by three orders of magnitude with latencies reaching up to 500ms and up to 1 in 1000 exceeding that range.

ETERNUS DSP OS Version 3.3.4 Media IOPs & Latency

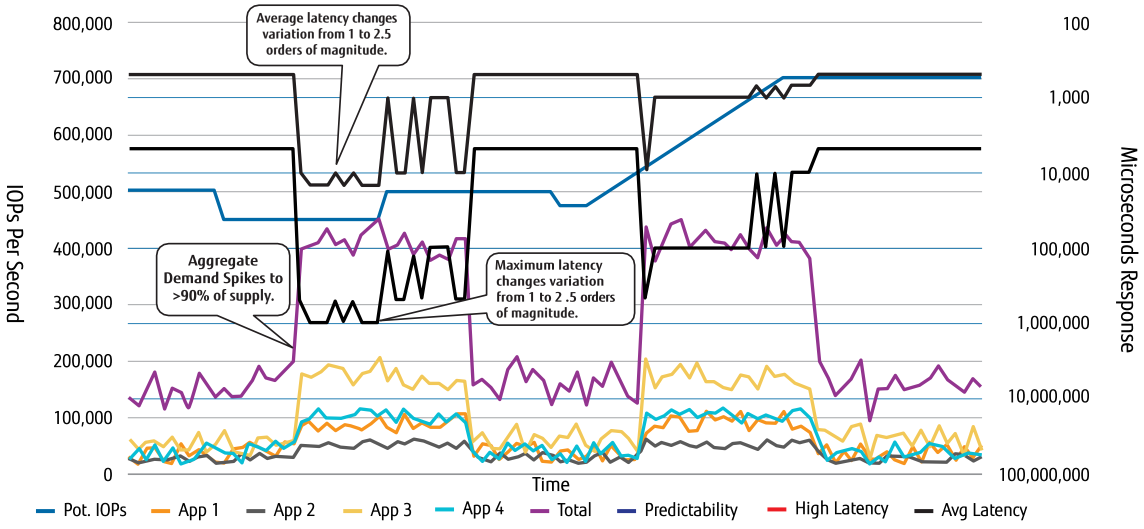


		Optane/Z-SSD	NVMe	SATA SSD	Hybrid
■	IOPS	250,000	150,000	125,000	100,000
▬	Latency µs (Average)	107	200	500	750

Figure 9-1 Per Node Measured Performance

What complicates the simplistic supply and demand view is that both supply and demand vary over time. At a given moment in time supply is fixed but there may be operational activities that consume non-trivial amounts of the available supply. Conversely, businesses, which drive application instances, have both cyclical and asynchronous behaviors. Most applications have cycles where their demand changes significantly. In addition, modern businesses seek to take advantage of opportunity and mitigate risk in real-time which in turn can create spikes in demand during these events. These dynamics represent the crux of the problem in talking about performance in the SDDC and how ETERNUS DSP was built for this environment.

Predictability as Supply / Demand Vary



The chart above depicts an environment with four applications running against a storage system. The predictability line is a log-scale depiction of predictability, measured in microseconds of average and long tail latency, as supply and demand vary over time. In this model, the applications have a dependent relationship such that all applications instances experience workload spikes at approximately the same time. While none of the applications demand is very high, as you can see from the graph, the aggregate demand approaches the available supply and the resulting predictability falls significantly as resource contention rises, particularly during the recovery from the media failure. This scenario is nearly a worst case scenario where all applications have increases in demand simultaneously and the system is recovering from a failure. Conversely, the same workload spike, after adding the additional node, reduced variability by nearly two orders of magnitude. While this is only a model of performance, it is quite frequently the case that demand increases for some business reason which then results in many applications having increased demand. Often applications for a pipeline of work so there may be a slight staggering of demand but aggregate demand spikes are not uncommon.

10. Putting it all together

In practice, how the system responds to variations, both on the supply side and the demand side, is the critical measure of long-term performance. Armed with this understanding, a system with the proper balance can be constructed, monitored and evolved over time that achieves consistent high performance as well as both higher order objectives -- business velocity and operational agility.

This model is designed to help enterprises model the performance that can be expected and the components needed to serve the multi-workload environment that is today's SDDC. And these measures are designed to be useful in sizing and scaling an overall system environment. Unlike historical performance measurements, whose goal it is to demonstrate what is possible under ideal (even contrived) conditions, the model described in this paper represents a modern data center in the real world, replete with many workloads and operational activities running simultaneously. The model also embraces the emerging customer metric of performance predictability and helps to represent the impact of supply / demand variation visually.

The downside of this model is that it exposes just how complicated a system servicing a multitude of disparate demands can be and how doing so may require additional resources to yield true performance predictability.
While it may not be good marketing, it is useful information for real world problems. For example, enabling deduplication in some leading systems (not the ETERNUS DSP platform) has been known to increase performance demand by as much as 50% with a commensurate impact on latency, IOPs and predictability. In an ETERNUS DSP system, deduplication only impacts supply as demand and supply converge, and the closed loop quality of service capability detects supply / demand convergence and triggers mechanism to reduce demand by deferring operational and data management activities. Once front-end application instance demand decreases, deferred operations catch up.

ETERNUS DSP's use of telemetry to detect increase in demand and using this information in a closed loop fashion to defer operations to ensure front-end demand is met within the SLA is a great example of ETERNUS DSP being built for real world events. Business needs vary over time, ETERNUS DSP was built for that, to enable velocity. Operational demands vary over time, ETERNUS DSP was built for that too – to enable agility.

All of us live in a world where data storage performance can be a key enabler (or limiter) in achieving business success. With the move to an SDDC, enterprises need a storage system built for velocity and agility, built for performance and scale and technology adoption and data management... the modern world of the SDDC is a world of and's, and ETERNUS DSP was built for the modern data center.

11. Conclusions

This paper introduces a model for evaluating performance of a SDDC data storage platform. Having such a model is essential in constructing, managing and evolving data storage and data management to help business achieve business velocity and operational agility.

Copyright Information

Fujitsu reserves the right to change any products described herein at any time, and without notice. Fujitsu assumes no responsibility or liability arising from the use of products described herein, except as expressly agreed to in writing by Fujitsu. The use or purchase of this product does not convey a license under any patent rights, trademark rights, or any other intellectual property rights of Fujitsu.

Contact
FUJITSU LIMITED
Website: https://www.fujitsu.com/eternus/

■ Trademarks
Third-party trademark information related to this product is available at:
https://www.fujitsu.com/global/products/computing/storage/eternus/trademarks.html

■ Disclaimer
All rights reserved, including intellectual property rights. Technical data subject to modifications and delivery subject to availability. Any liability that the data and illustrations are complete, actual or correct is excluded. FUJITSU LIMITED is not responsible for any damage or indemnity that might be caused by the content in this document.