An Oracle White Paper
September 2011

Delivering Quality of Service with Pillar Axiom 600

Executive Summary 1
Dynamically Controlled Quality of Service (QoS) in Pillar Axiom 600 Systems 1
Defining the RIght Storage Problems to Address 2
Storage Multi-Tenancy: The Good, the Bad, and the Ugly 2
What Has Value to the Business? 2
Problems Solved: Pillar Axiom 600 with Quality of Service 3
Resource Allocation 3
Prioritizing Access to Storage Resources with Quality of Service 3
Pillar Axiom 600 Architectural Overview 5
Optimizing for a Specific Workload 7
Application Optimization by Tuning the Entire System 8
Application Aware Profiles 9
Use Case #1: Oracle Automatic Storage Management (ASM) 9
Use Case #2: Microsoft Exchange 10
Conclusion 11

Executive Summary

The Pillar Axiom 600 system was designed from the ground up to deliver business-critical, deterministic Quality of Service (QoS) consistent with the business value of the application and its data. It eliminates the requirement to overbuy and underutilize storage to buffer against application interference in a multi-tenancy shared storage system. This allows organizations to buy the capacity they need, scale when needed, and conserve capital in the process—all while delivering consistent performance under load.

Storage administrators have been dealing with storage performance or QoS issues since disk drives were first commercialized by IBM. This problem has worsened over time as processors have gotten faster and disks have simply not kept up. There is now a difference of about six orders of magnitude in processor cycle times versus disk I/O times. Disks operate in milliseconds whereas processors operate in nanoseconds. This has led to an inability to effectively use the growing disk drive capacity since drives tend to run out of IOPS long before being filled to capacity. This forces the purchase of extra disks to ensure that applications are getting the I/O response times required to meet business needs.

Overbuying storage capacity to get performance is very poor economics. In addition to the extra cost of unused storage capacity, ongoing expenses (OPEX) tend to be higher because a over provisioned system takes more time to manage and has higher power and cooling costs.

Dynamically Controlled Quality of Service (QoS) in Pillar Axiom 600 Systems

To effectively manage storage costs requires a new approach in which QoS is dynamically managed to meet application needs. Most of today's storage controllers were never engineered to enable QoS to be controlled under load. As a result, these designs cannot dynamically manage QoS to ensure that critical business applications receive the appropriate level of I/O performance regardless of other activity on the storage system. Oracle's Pillar Axiom 600 system addresses this problem by prioritizing I/O requests based on business requirements for performance based on the value of the application. The core data-path infrastructure of the Pillar Axiom 600 system thus uses its integrated, dynamic management to optimize the business value of the application environment.

Defining the Right Storage Problems to Address

Storage Multi-Tenancy: The Good, the Bad, and the Ugly

Storage multi-tenancy has been driven by three key factors:

- Storage has moved from a dedicated, direct attached resource to a completely shared resource, either via a SAN (in the case of block-mode storage) or an IP-based network (in the case of NAS), because customers wanted to consolidate storage to optimize both management and storage utilization. In business terms, customers wanted to reduce both CAPEX and OPEX costs.
- The dramatically increased capacity of disk storage today typically means that data from multiple applications can easily reside on a single storage system.
- The increased use of scale-out virtualization is driving the acceptance and value of multi-tenancy, virtualized shared storage.

Unfortunately, this concept of a shared resource and consolidation causes other storage management issues. Specifically, once data from multiple applications is stored and managed on the same device, the I/O workloads can't help but have the potential to interfere with each other. A large database scan surely will interfere with short, bursty, latency-sensitive transaction data. As is typical with the design of most storage devices, a large number of disks are controlled by a small number of RAID controllers, a situation that can, and often does, cause a bottleneck. Streaming large block I/Os that are not going to be reused can quickly pollute a controller's cache and force a de-stage of transaction data that has a high probability of reuse. Addressing this problem by separating these two types of data onto different storage devices is directly contrary to storage consolidation and results in increased management overhead and, most likely, lowered utilization of the two storage systems.

What Has Value to the Business?

When talking about investing, one speaks about return on investment (ROI), competitive positioning, and total cost of ownership. IT spending, however, is really just an expense—pure and simple. It's there solely to support the one thing that really does have value: the application and its data. Oracle recognizes the fact that the application is what has real value and all the other components of the IT infrastructure (servers, networking gear, and storage) are there only to support the application. Given this, it only makes sense to design a storage system that can manage and prioritize its resources based on the business value of any given application versus just responding to all I/O requests equally.

For example, an order entry application would typically be considered business-critical since the ability to take and process orders is a direct revenue generation function. Certain HR applications, on the other hand, may not be so critical. Slower I/O response time may not necessarily be desired for the HR application, but the business impact is surely less than a potential customer abandoning an order because the response time was too long.

The fundamental design point of the Pillar Axiom 600 is to prioritize all of the storage system's resources based on an application's value. In other words, for the first time a storage device can be made to be fully *application-aware*.

Problems Solved: Pillar Axiom 600 with Quality of Service

Resource Allocation

The Pillar Axiom 600 is specifically designed to dynamically allocate all the resources of the storage system in the data path (CPU, cache, capacity, bandwidth, I/O queuing priority, RAID type, and physical data placement on the optimal type of disk device). These systems ensure that, even under very heavy load situations, applications with high business value will always get sufficient resource to deliver the required QoS (that is, performance) that can meet or exceed the defined business requirements.

The Pillar Axiom 600 is the first offering to break with the simplistic FIFO command management model that has been a staple of the industry for the past 40 years. Pillar Axiom 600 storage systems move to a model that allocates storage system resource based on the value of the application or data, not just based on which application got an I/O command into the system first.

Prioritizing Access to Storage Resources with Quality of Service

Many of today's storage environments service multiple applications in a fully shared, multi-tenancy model. This centralized approach simplifies management and offers economies of scale in a consolidated storage infrastructure. However, problems can arise when one application places a heavy load on the system to the detriment of other applications that may be trying to access their data at the same time. Typical block-mode storage devices have no awareness of what application is actually issuing the I/O and whether that application is important to the business or not. They use a simple first-in, first-out (FIFO) command management model to allocate disk accesses, controller bandwidth, controller CPU cycles, and command queuing priorities. Said differently, there has been no way—other than via manual intervention by moving data to different drives or arrays—to control how much and when any given application uses storage resources.

Recently, several vendors have introduced sub LUN automatic tiering where frequently referenced blocks of data are promoted to faster media and less frequently referenced blocks are demoted to slower, less expensive media. While this sounds like a perfect solution to delivering performance where and when needed, it has one fatal flaw: Historical frequency of access may have absolutely no relationship to the business value of the application. Looking backwards is not necessarily a good way to predict the future. Additionally, reaction times for these automatic block migration algorithms are typically very slow. In one vendors case it "thinks" about promoting data blocks for three days before it does it, and then "thinks" for 12 days before demoting data. Normally, when you have a performance problem, cogitating for 72 hours before you address it is not acceptable.

With Pillar Axiom QoS you never have to worry about this. You know going in that your application is going to receive deterministic performance based on its business value regardless of system load with no "cogitate" time based only on looking in the rear view mirror.

To deal with the inability of disk storage systems to manage QoS, many organizations have traditionally over-provisioned (that is, overbought) storage resources to ensure that there would not be any I/O contention. This is simply terrible economics. One of the most common rules of thumb was to over-provision I/O bandwidth by 40 percent and IOPS by 50 percent to statistically reduce the chance of hot spots unexpectedly forming due to I/O inter-arrival rate skew[1]. One can either define this result as achieving low utilization of the storage device or paying for additional equipment that can't be used. Either way it leads to the same thing: achieving stable QoS under load becomes very expensive, approaching almost twice the optimal cost.

The recent trend to move to the virtualized scale-out model of computing has exacerbated this problem. Rather than a few large servers accessing storage, many (hundreds to thousands) of smaller servers, often with multiple O/S images on them, are all now contending with each other for access to disk. This chaotic workload is neither stable nor predictable; rather it generates highly bursty I/O requests that can change rapidly over time. As load increases, queuing theory teaches us, even small perturbations in I/O skew rate can have dramatic effects on response time[2]. If you commute daily on a heavily used freeway, you probably already have an implicit understanding of this concept.

Dedicated, non-shared direct attached storage in a virtualized environment is generally a contradiction in terms. The resource allocation flexibility that is one of the prime reasons to virtualize resources is destroyed, as is the high utilization model. External, public, virtualized compute facilities (known as *clouds*) have this problem in spades. Users expect a certain QoS for which they pay a fee. The cloud operator has precious little control over what any given user's application suite does at any given time, yet is somehow expected to ensure that everybody gets the QoS they paid for. The cloud operator is faced with a Hobson's choice. He or she can either over-provision the cloud storage, and possibly also the compute resource, and end up making things so expensive that the service is not competitive, or reduce I/O usage charges without guaranteeing QoS, running the risk of poor customer satisfaction and potential lost business.

Everything stated so far also applies to server provisioning as well, the difference being that the server operating system (OS) resource management tools are far more sophisticated than their storage brethren. Server OS resource managers are highly application-aware since that is the entity that they are controlling. Block-mode storage devices, on the other hand, have absolutely no knowledge of applications or of their relative value to the business.

[1] Inter-arrival rate skew represents the fact that I/Os are typically bursty. A greater number of disparate applications sharing any given storage resource will tend to exacerbate the effects of I/O inter-arrival skew.

[2] While this seems to contradict Little's Law, Markov (memory less exponentially distributed inter-arrival rates) tends to negate the simplistic "Black Box" service time functions implicit to Little's constant service time model. Little's Law assumes a non-shared, non-interfering queue server process. This is not the case with generic FIFO based storage systems dealing with interfering workloads.

Pillar Axiom 600 Architectural Overview

Oracle has addressed this problem by designing the Pillar Axiom 600 solution around two key principles:

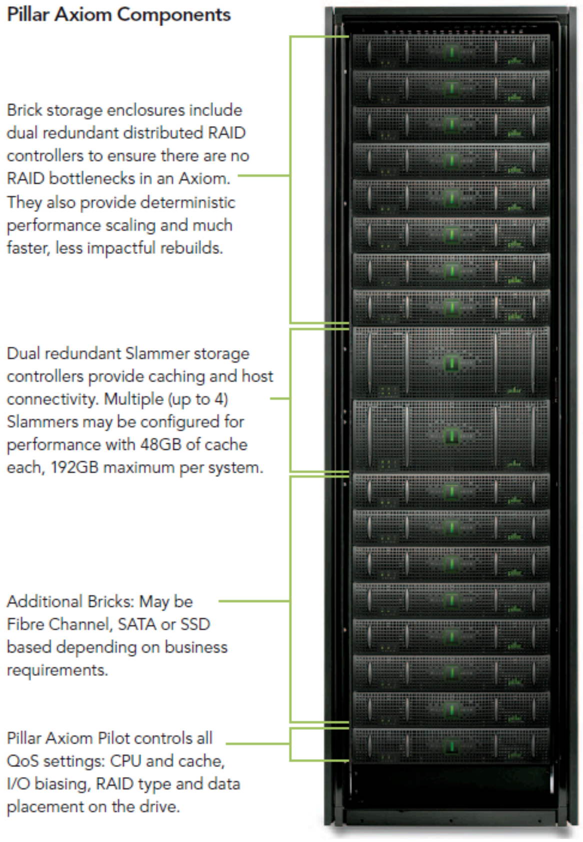
1. Identify all choke points such as RAID controllers, cache, and internal paths and eliminate them.
2. Allow the user to specify by application the desired QoS level required to meet the business needs and have this QoS-level policy control the allocation of all storage system resources in the data path to meet the business value of the application.

Choke point elimination in a Pillar Axiom 600 system is accomplished by implementing a distributed RAID design instead of a single or dual controller choke point design. Every Pillar Axiom Brick (storage enclosure) in a Pillar Axiom 600 system has its own dual redundant set of RAID controllers, each with its own half gigabyte of cache memory. This spreads the overhead of performing read-modify writes, in the case of RAID 5, across a large number of RAID controllers versus just one or two controllers. For example, a large Pillar Axiom 600 system with 24 disk Bricks would have 48 individual RAID controllers, all of which can operate in parallel.

Sitting above the distributed RAID controllers are one or more dual-redundant storage controllers (Pillar Axiom Slammer control units) that provide 48 GB of cache per Slammer and implement the application-aware QoS resource prioritization discussed earlier. With 48 GB of cache per Slammer, the Pillar Axiom 600 provides enough cache so that applications with differing I/O patterns will not crowd each other out. With up to four Slammers per system, Pillar Axiom 600 can support as much as 192 GB of write protected cache.

In addition, an application with a high QoS policy will be guaranteed a certain percentage of controller CPU cycles regardless of the load on the system. In cases where there may be an extreme load on the system, and the I/O patterns of different applications are mutually antagonistic to each other, application I/O may be segregated to different Slammers to ensure that there is absolutely no chance of any interference with each other. An example of this might be in health care, where accessing very large 265 slice MRI images might conflict with routine patient admitting. The MRI images could be accessed through a different Slammer than other general hospital transactional applications, thus guaranteeing completely stable QoS under any load. Proper system configuration can ensure that there are virtually no choke points in a Pillar Axiom 600 system. Even in a single Slammer configuration, antagonistic I/Os can be segregated to one of the dual controllers in each Slammer so as to reduce interference dramatically.

In addition, the Pillar Axiom supports Storage Domains that can physically isolate specific VLUNS to specific Bricks (Storage Enclosures) to absolutely ensure that there is no inter-application I/O interference, thus guaranteeing perfectly deterministic performance regardless of other activity in the array.



Pillar Axiom Components

Brick storage enclosures include dual redundant distributed RAID controllers to ensure there are no RAID bottlenecks in an Axiom. They also provide deterministic performance scaling and much faster, less impactful rebuilds.

Dual redundant Slammer storage controllers provide caching and host connectivity. Multiple (up to 4) Slammers may be configured for performance with 48GB of cache each, 192GB maximum per system.

Additional Bricks: May be Fibre Channel, SATA or SSD based depending on business requirements.

Pillar Axiom Pilot controls all QoS settings: CPU and cache, I/O biasing, RAID type and data placement on the drive.

Figure 1. The components in the Pillar Axiom 600 storage system

It may not always be possible or economically viable to completely segregate differing workloads from each other. To address this issue, the QoS policies within each Slammer, managed by the Pillar Dynamic Performance Manager, can control the allocation of available CPU cycles in a Slammer to designated applications, thus prioritizing the order in which I/O commands are processed. High priority applications will receive a guaranteed percentage of available controller CPU cycles under any load condition as well as have their I/O requests processed at a higher priority than applications with a lower QoS setting.

A side benefit of this prioritization is that high priority applications implicitly get more cache space as their I/O activity rate ensures that cached data is not flushed to disk when the cache is under load. The combination of these various policies virtually guarantees that business-critical applications will continue to receive outstanding performance under any load and any application mix. In addition, Pillar Axiom 600's QoS polices allow the user to specify to the system the expected I/O patterns of each application (random, sequential, read or write-biased, and mixed). Oracle will dynamically optimize both RAID type, physical data placement on the optimal drive type (SSD, FC, or SATA), and cache read-ahead aggressiveness so as to optimally use all system resources in the most efficient way.

Optimizing for a Specific Workload

Most other storage systems just allow a base set of parameters to be established at installation time and these are typically compromise settings that are not optimized to any given application; rather they represent a lowest common denominator approach to system optimization. Stated another way, the storage system is not set optimally for any specific workload nor is it going to be really terrible at any one specific workload.

Because workloads and priorities can and do change, all QoS policies can be modified at any time to reflect new business conditions. Application QoS can be increased or decreased as appropriate. For instance, quarter-end reporting might require a temporary increase in QoS settings for an application. Unlike most other storage systems, Pillar Axiom 600 data prioritization is dynamic, not static. The compromise settings discussed above tend to be permanent and are made at system installation time. They cannot be dynamically adjusted to match changing business priorities the way Pillar Axiom 600 can.

Pillar Axiom 600 systems currently support three classes of drives:

- High performance fibre channel 15K RPM drives
- Large capacity, low cost SATA drives
- Extreme performance solid-state drives (SSDs) where seek and latency are virtually eliminated

While a large capacity SATA drive may be perfectly acceptable for infrequently referenced archival data, disk-to-disk backups, or large block streaming data, they are far too slow for most transactional systems. As part of the entire QoS strategy, optimal data placement on the appropriate type and number of drives is crucial to round out the entire QoS picture. What the QoS manager does is interpret the QoS policy settings and make decisions on where to physically place data based on both the QoS service level requested as well as what actual devices are installed on the system. For example, for a system with both fibre channel high performance drives and SATA drives, Pillar Axiom 600 would treat them as two separate "storage classes," each with access to all five QoS levels. Premium QoS data (the highest level) would be striped across up to 48 drives, thus delivering the ability to accept very high IOPS or bandwidth rates to any given LUN or set of LUNs. [3]. Archival data (the lowest QoS level) would be striped across fewer disks since the performance level being requested is lower[4]. With Pillar Axiom 600, volumes may be thinly provisioned so only the space that is actually

[3] Starting with firmware release 4.1, SSD's FC and SATA disks may all concurrently reside in a single Axiom system. Each disk type "Storage Class" now can be managed with access to all five QoS levels. User policies control which class data is selected when LUNs are created and how many drives are striped.

[4] It is not required to stripe across all media. Typically, in the case of other vendor's arrays, stripe sets cross all media of the same type. The effect of this is to load balance I/O across all similar disks in an attempt reduce or eliminate hot spots seen on high activity traditional LUNs. The problem here is that inter-application I/O interference is exacerbated. Pillar allows you to define stripe sets by application and

used is physically allocated. This saves money and can also reduce seek times and improve performance overall as there will be fewer tracks to cross in most cases because the VLUNs are usually not completely filled.

The previous discussion about QoS settings explained that they could be dynamically changed if business requirements change. This can be done in two ways: temporarily or permanently. A temporary change will change the CPU, cache, and I/O priority settings, but will not actually move data to a new storage class. A temporary setting is typically used if there is a need for more performance for a short period of time, such as during a quarter-end reporting period. A permanent change may physically migrate the data to a different drive type, if so requested. The new data location would match the selected QoS storage class settings. RAID types may also change if the new QoS settings specify a different I/O workload mix. For instance, if the old I/O workload setting were random write, the underlying RAID type would have been RAID 10. If the new setting were mixed, the new RAID type would be RAID 50. This data movement takes place in the background, and the data is still available during the migration. QoS policies in the Pillar Axiom 600 system make sure that the overhead of internal data migration is managed in a way that does not impact high QoS application performance.

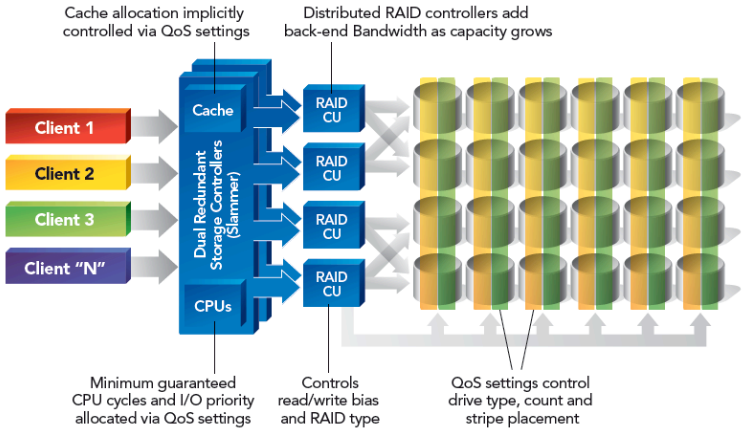

Cache allocation implicitly controlled via QoS settings
Distributed RAID controllers add back-end Bandwidth as capacity grows
Client 1
Client 2
Client 3
Client "N"
Cache
Dual Redundant Storage Controllers (Slammer)
CPUs
RAID CU
RAID CU
RAID CU
RAID CU
Minimum guaranteed CPU cycles and I/O priority allocated via QoS settings
Controls read/write bias and RAID type
QoS settings control drive type, count and stripe placement

Figure 2. QoS resource allocation control points.

Application Optimization by Tuning the Entire System

The entire data path through the storage system must be managed, right down to physical placement of data on different disk types in order to deliver true QoS. Figures 2 and 3 depict how Oracle's QoS manages the resources of a complete Pillar Axiom 600 storage system.

minimize inter-application interference. With the Pillar Axiom, disk volumes, or file systems in the case of NAS, are made on Virtual LUNs (VLUNs) that are then dynamically mapped to physical Stripe Groups under QoS physical placement control.

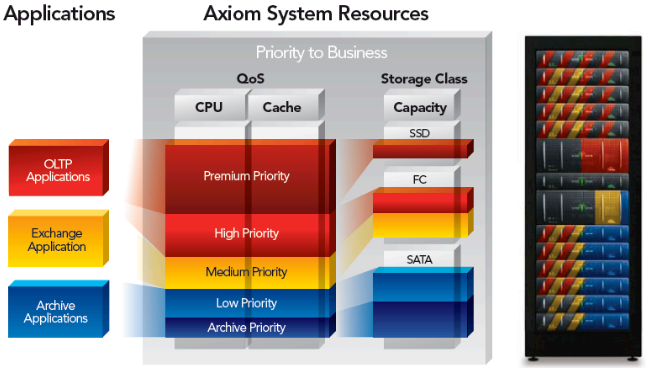


Figure 3. Tuning the entire system with QoS controlling the cache, CPU, and capacity.

Application Aware Profiles

While it is possible to set all QoS parameters to exactly match a given workload, to do so presumes a very clear and detailed understanding of the I/O workload that the applications generate. In cases where this detailed information may not be well characterized, Oracle has created predefined profiles that are optimized to many common workloads. An application profile is a predefined set of QoS policy settings that have been tested and proven to work well with various applications. Instead of having to select all the settings that have been described in this paper, administrators can simply select a profile from the Pillar Axiom 600's management GUI and the system takes care of the rest. Administrators can even define their own unique profiles to make repetitive storage provisioning much faster and perfectly optimized to a specific workload.

Examples of common application profiles are: Oracle 11g Transaction Processing, Automatic Storage Management (ASM), Table and Index space, Redo and Archive logs, Microsoft Exchange, VMware, Disk-to-Disk VTL backup, and Web-facing files.

The following sections describe two examples of applications where Pillar Axiom 600's QoS management can be crucial to achieving business-critical stability and guaranteed performance.

Use Case #1: Oracle Automatic Storage Management (ASM)

Pillar Axiom 600 has specifically optimized the Oracle Application Profile to match the I/O patterns that are generated when ASM is being used. ASM's normal I/O pattern is to write and read 1MB random I/Os to disk. This I/O size is not optimal for achieving high bandwidth with most internal RAID controller stripe settings. A special deep stripe that is exactly 1MB in size allows Oracle ASM I/Os to require fewer internal writes to disk. This results in dramatically improved read performance with lower disk overhead and higher overall sustained bandwidth. In addition, Pillar Axiom 600's large cache in write-back mode will attempt to buffer multiple writes and write out complete RAID stripes when possible.

Based on the overall Oracle ASM performance required, other Pillar Axiom 600 settings can be adjusted. Application QoS can be set *high* to ensure that there are guaranteed CPU allocations and that disk placement is favorable. In a moderately active system, lower QoS settings can be chosen that may allow the use of less expensive SATA disks, while still delivering the business-required QoS levels. Again, should more resources be needed, the application QoS priority settings can be raised: either on a temporary basis or permanently.

Use Case #2: Microsoft Exchange

Email has become more than just a method to communicate. It has become an integral part of the business process in many companies, and as such, is considered a business-critical process. Microsoft Exchange is also very sensitive to I/O latency—especially write latency. It tends to generate a lot of short, bursty I/Os that need to be serviced quickly and consistently. In the past, most customers have dedicated direct attached storage to Microsoft Exchange to make sure that other applications would not contend with Microsoft Exchange's need to have low latency response time. Microsoft actually recommended this configuration in order to avoid customer satisfaction problems. This, of course, led to increased management issues with multiple storage silos. In some cases, it also resulted in poor utilization as the storage was over-provisioned to ensure that there would never be a short storage situation or poor response times because the physical disks had reached their limit of sustainable IOPS Both the short on storage situation and going beyond a sustainable IOPS range can lead to a crash of Microsoft Exchange. To reduce this risk, many customers were paying too much for storage in order to increase the stability of Exchange.

With Pillar Axiom 600, a completely different paradigm comes into play. Starting at the very top, to optimize storage for Microsoft Exchange, the I/O workload can be set to *random write biased*, which creates a RAID 10 stripe set. This is crucial, as RAID 5 in degraded mode simply cannot deliver the I/O rate that a reasonably sized Microsoft Exchange system requires for stability. This also ensures that cache write-back buffering is available to handle bursty I/O peaks and ensures very low latency writes. Secondly, the application priority can be set to either *premium*, in the case of a really large installation under very heavy load, or *high* for a less busy system. Both of these settings guarantee that a minimum amount of CPU cycles will be available for Microsoft Exchange regardless of what other applications may be doing. Both of these settings tend to favor the higher performance outer bands of the disk. Keep in mind that if the Microsoft Exchange installation grows rapidly, a *high* application priority can be easily changed to *premium*. Furthermore, with thin provisioning, more capacity can be dynamically added if and when required.

The result of these QoS settings, along with Pillar Axiom 600's distributed RAID architecture, is that Microsoft Exchange now has exactly the amount of storage that it really needs due to Pillar Axiom 600's thin provisioning. Interference from other applications has been fenced off so that Microsoft Exchange data can safely reside on shared multi-tenancy storage. Pillar Axiom 600's large cache and guaranteed CPU allocation along with I/O priority and implicit cache allocation delivers sufficient headroom so that the physical disks do not have to shoulder the entire load and more of the disk capacity can safely be used.

The net result of this installation is that customers get a very stable, future-proofed Microsoft Exchange installation at a lower acquisition cost than if they had over-bought direct attached storage.

Conclusion

The Pillar Axiom 600 system was designed from the ground up to deliver business-critical, guaranteed QoS consistent with the business value of the application and its data. It eliminates the requirement to overbuy and underutilize storage to buffer against application interference in a multi-tenancy shared storage system. This allows organizations to buy the capacity they need, scale needed, and conserve capital in the process—all while delivering consistent performance under load.

Oracle is committed to developing practices and products that help protect the environment

Delivering Quality of Service on Shared Storage
with Pillar Axiom 600
September 2011

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200

oracle.com

Hardware and Software, Engineered to Work Together