

# Storage Platform Assessment Service Recommended Practices for Optimizing Storage Performance

By Jack Moreland  
*Solutions Architect*  
*Hitachi Data Systems*  
*Global Solution Services*

March 2007

## Table of Contents

|   |    |
|---|----|
| 1. Introduction.....                            | 3  |
| 1.1. Tools.....                                 | 3  |
| 2. Accepted Practices.....                      | 4  |
| 2.1. General Practices.....                     | 4  |
| 2.2. Performance-related Items.....             | 5  |
| 2.2.1. Quick Hits.....                          | 5  |
| 2.2.2. Port Utilization.....                    | 5  |
| 2.2.3. ACP Utilization.....                     | 6  |
| 2.2.4. Cache Utilization.....                   | 7  |
| 2.2.5. Array Group Utilization.....             | 7  |
| 2.3. Known Bad Practices.....                   | 8  |
| 2.3.1. I/O Load concentration.....              | 8  |
| 2.3.2. "Anywhere it can be Found".....          | 8  |
| 2.3.3. Overly dispersed.....                    | 8  |
| 2.3.4. Workload Type Versus RAID Types.....     | 8  |
| 2.3.5. Uneven Workload Distribution.....        | 8  |
| 2.3.6. Array Groups to Host Segregation.....    | 8  |
| 3. Storage Pool Discussion.....                 | 9  |
| 3.1. Implement Storage Pools.....               | 9  |
| 3.2. Standardize Emulation Types and Sizes..... | 10 |
| 3.3. RAID Types.....                            | 11 |
| 3.4. Other Considerations.....                  | 11 |
| 4. Summary.....                                 | 12 |

# 1. Introduction

Hitachi Data Systems Global Solution Services (GSS) offers a Storage Platform Assessment service that provides customers with a health check and performance evaluation of their storage infrastructure. This whitepaper summarizes the key areas that are evaluated in that service and some recommended best practices for optimizing storage performance. It highlights some of the most common problem areas and suggestions on how to avoid them.

This whitepaper is targeted to storage administrators and assumes an understanding of the architecture, and a working knowledge, of Hitachi's enterprise storage systems. It is neither a cookbook nor a tutorial on storage performance management. Rather it is intended as a guide for fine tuning an existing storage infrastructure. This whitepaper simply provides some considerations and highlights some potential bottlenecks as observed at some storage environments.

The following sections were compiled from Storage Platform Assessment reports created for customers who were experiencing performance problems. They provide some insights into generally accepted practices, performance considerations, and a discussion on the concept of storage pools.

## **1.1. *Tools***

The Storage Platform Assessment service makes extensive use of Hitachi Tuning Manager® software, and additional software tools to investigate performance problems. As such it should be noted that port (front-end or back-end) microprocessor utilization can only be measured for the entire port. It cannot be broken down per server that consumed the cycles. However, IOs per second (IOPS) and megabytes per second (MB/s) can be measured per individual logical device (LDEV), and by any aggregation of LDEVs for which there is a configuration data relationship such as Array Groups, Ports, Owners, Products (for example True Copy software and Hitachi ShadowImage Heterogeneous In-System Replication software).

NOTE: Programs functioning as IO workload generators, such as IO-meter, even with basic parameter settings, are capable of unleashing a stream of I/O that is far more intense than most applications ever produce. A single small Microsoft Windows server can crush a port or an array group of any storage system. If you happen to crush an array group with writes, it causes high write pending and when write pending gets to 70%, the storage system, in an act of self defense, stops accepting writes from every attached server until it can make more room in cache via de-staging. In a Universal Storage Platform, this impact is limited to a cache partition.

## 2. Accepted Practices

The Accepted Practices are broken down into General Practices, Performance Considerations, and Known Bad Practices. These practices, rules of thumb, and insights are provided as a guide and may not apply in all situations. Always try to get as much data and information as possible in order to make an informed decision.

### 2.1. *General Practices*

- Storage is a shared resource, similar to network bandwidth. It is a recommended general practice to carefully consider the impact of short-term, intense workloads such as database loads/refreshes before starting such workloads. Furthermore implementation and adherence to operational change control procedures to manage these types of events is highly recommended. Change control procedures, for example, may alert the storage administrators in advance to such activities as database loads, allowing them to make appropriate business decisions to schedule these activities and/or to ensure that these activities make the most efficient use of available resources.
- When assigning workload to ports and/or array groups, the I/O profile of hosts sharing the resource should be considered. In particular, the I/O size for a host is very important. Front-end port pairs perform the best with a uniform physical IO size, as opposed to a disparate mix of I/O sizes.

Unfortunately, we have no direct measure of IO size distributions within the Hitachi equipment and using average IO sizes as a basis of analysis can conceal the reality of mixed IO sizes.

- It is a good practice to standardize on the fewest different combinations technology and LDEV sizes required to satisfy specific application requirements. By minimizing the number of emulation and RAID types in their configuration storage administrators can standardize their storage, thereby making their storage inventory easier to manage.
- The storage deployment should be engineered, to the extent possible, to provide every server with more bandwidth between disk and cache than is provided by the front end ports. This may be accomplished by:
  - Choosing the correct RAID type for the I/O profile
  - Providing enough disks to service the workload with techniques like interleaved storage pools
  - Automating the distribution of I/O traffic with host logical volume striping
  - Ensuring that the manual distribution of traffic across array groups is as even as possible across all of the disks made available to the server
- It is not necessary to disperse the I/O of every application in the pool across every array group in the pool. In fact it is common to define subsets of the pool as I/O dispersal groups. Nonetheless, it is generally beneficial to distribute the I/O of each application across more than the minimum number of array groups required to provide the storage space. This in turn means that applications typically engage in managed sharing of array group resources.

## **2.2. Performance-related Items**

### **2.2.1. Quick Hits**

- When the cache write pending percentage reaches 70%, a Lightning 9980V system will stop accepting new writes in an attempt to destage the writes currently in cache. This type of spike has a severe impact on all hosts using the Lightning 9980V system.
- It is also worth noting that the array control processors (ACPs) in a Universal Storage Platform can handle twice the workload of the Lightning 9980V system ACPs.
- The recommended rule of thumb for planning port pair capacity; one that allows enough room for growth and unplanned load spikes is to stay within 2500-3000 IOPS or 100MB/sec-120MB/sec per port pair, whichever is encountered first. This rule also allows for one member of a port pair to fail, and the survivor to carry the entire load.

### **2.2.2. Port Utilization**

Storage administrators should work to balance a system or predict new loads, MB/s and IOPS are typically used to plan load assignments because port pair microprocessor requirements are unknown for new loads. Actual port pair capacity depends on the I/O profile of the applications using the port pair, and in fact, conflicting I/O profiles can reduce this capacity. A useful rule-of-thumb for planning front-end port pair capacity is 2500-3000 IOPS or 100MB/sec-120 MB/sec, whichever is encountered first. The actual port capacity will vary depending upon the profiles of the specific workloads presented to the port by the servers, hence it is always important to monitor port microprocessor utilization after implementation, and make adjustments if necessary.

It should be noted that reaching 100% utilization may not be possible due to the burst profile of hosts. Planning for 70%-80% as the maximum available utilization for any resource is recommended. Consequently, it is good practice to plan for 35% to 50% utilization on a resource pair and approximately 60% utilization for microprocessor resources deployed in active-active quads.

In order to achieve optimal utilization of the front-end port pairs, the I/O load should be evenly distributed across all port pairs as much as practical. This is relatively easy to implement and can have immediate benefits. Several recommended practices exist to guide this effort:

- I/O profiles should play a roll in the grouping process
  - Group predictable and compatible I/O profiles
  - Attempt to separate large block I/O and small block I/O onto different port pairs when possible
  - Group sporadic I/O profiles
- Consider redundancy requirements
  - Avoid introducing a single point of failure
  - Failover should not overwhelm a port or port pair, that is, avoid creating a cascading failure
- Service levels should play a role in the grouping process
  - Physically or Logically isolate critical hosts

High microprocessor utilization is an unambiguous indication of high port utilization. However, low microprocessor utilization is not by itself a definitive indication of low port utilization. When port microprocessor utilization is low, throughput in MB/s must also be examined before concluding that port utilization is low.

Throughput constraints for small block I/O traffic, less than 64K in size, will typically manifest themselves as high microprocessor utilization, while port throughput constraints for large block I/Os manifest

themselves as high Fibre Channel utilization, for example, MB/s. Thus, block size must be considered when choosing a metric to evaluate port utilization.

High port utilization is not necessarily a problem. Some applications are deployed with a design goal of maximizing throughput and their demand is only limited, intentionally limited, by full utilization of one or more storage resources. Response time sensitive applications generally avoid high resource utilization because these high levels of utilization contribute to extended response times.

Another factor to consider in planning port loading is what happens when a port fails. If ports are deployed in pairs, then the port microprocessor utilization of each member of the pair should be kept below approximately 40% during the normal load cycle to allow continued operations without degradation in the event that one member of the pair fails, and the surviving port must carry the entire load (now 80%). . In cases where port pair microprocessor utilization is above 80% it is likely that hosts are negatively impacting each other as they compete for storage processing resources.

One way to improve on the effective capacity of ports is to deploy I/O paths in groups of four, also known as quads. If one path of a quad fails, then 25% of the capacity available to the affected servers is lost. However, if one path of a pair fails, then 50% of the capacity available to the affected servers is lost.

The net of this failure consideration is, assuming 80% utilization represents effective full utilization, and assuming continued operation without degradation is an objective:

- 40% should be considered full utilization under normal operating circumstances for port pairs and ACP pairs
- 60% should be considered full utilization under normal operating circumstances for paths in a balanced port quad

Server Priority Manager software allows the storage administrator to decide how much of a port's resources a given host is allowed to use. This can help mitigate cases where one host unexpectedly attempts to consume resources on a shared port pair. The centralization and flexibility of Server Priority Manager is easier to manage than host based SCSI throttle settings. Server Priority Manager can also be implemented and modified without the need to reboot the hosts.

### **2.2.3. ACP Utilization**

At many customer sites ACP pairs utilization rates have been observed at up to 80%. It is recommended that the utilization of an ACP be kept in the 35%-50% range. The reason for this recommendation is that a failure on a single ACP means the remaining ACP in that pair would be responsible for all the work of the pair. If the workload for the pair is above 50% the remaining ACP would not be able to service the entire workload causing IO degradation. The higher above 50% an ACP pair workload the larger the degradation would be in the event of an ACP failure. The higher above 50% an ACP pair workload moves, the larger the degradation becomes. There are three (3) options to alleviate ACP over utilization:

- One option is to add more ACPs and disk enclosures, then to evenly distribute the disks (array groups) across all ACP pairs in the storage system. This allows the ACP utilization to remain within recommended deployment practices and ultimately ensure the storage system is not susceptible to I/O degradation caused by single-point failures.
- Another option to prevent ACP over-utilization would be to migrate to a Universal Storage Platform as the ACPs in a Universal Storage Platform can handle twice the load of a Lightning 9980V system.
- The last option would be to mitigate the spikes in ACP utilization to the recommended 35%- 50% range.

#### **2.2.4. Cache Utilization**

A primary concern is the cache write pending rate. When the rate of data transfer from hosts into cache exceeds the rate of data transfer rate from cache to disk, the writes accumulate in cache. The metric ‘% write pending’ is the percentage of cache occupied by writes that have yet to be de-staged to disk and is a measure of this accumulation of write data in cache. Writes pending reaching 70% can severely impact all applications using the Lightning 9980V system.

Write Pending of 30% or below is considered normal operation. Write pending of 40% or above warrants corrective action. If a write pending reaches 70% the Lightning 9980V system stops accepting new writes as it attempts to de-stage the writes currently in cache. This condition is disruptive to all servers attached to the storage and should not be allowed to occur and especially during normal production operations.

#### **2.2.5. Array Group Utilization**

The key metrics when evaluating Array Group performance are ‘tracks/sec’ and ‘% utilization’.

- An array group with a high number of tracks per second could be a sign of over utilization of that array group. If an array group is over-utilized it can cause degradation of I/O performance to the host on reads and data backing up into cache on writes.
- An array group with utilization above 50% is a cause for concern. In order to allow enough reserve capacity for prompt failure recovery (sparing) concurrent with ongoing operations. Like microprocessors, array groups typically run out of available capacity between 70% and 100% utilization, exactly where is a consequence of the application(s) burst profile.

The use of a host level Logical Volume Manager (LVM) Striping and Storage Pools, both discussed later in this document, can help to alleviate single array groups having a high tracks/sec by distributing that I/O across a larger number of array groups.

To Implement LVM, four LDEVs are identified—one each from an Array Group in each ACP pair. The Array Groups that the LDEVs come from should have matching attributes, and the LDEVs should come from similar positions on the disk. The four LDEVs are then presented to the host and striped together using a logical volume manager. The result is a host volume that is evenly distributed across multiple Array Groups and multiple ACP pairs. LVM striping will help to balance the I/O from any host across all four ACP pairs and mitigate the possibility of any one Array Group becoming a throughput constraint. The number four is a commonly used number. Nonetheless, different circumstances may dictate a different number of stripe columns.

Furthermore, striping increases the back-end bandwidth available to a host volume. If the back-end bandwidth provided is greater than the front-end bandwidth provided, then a buildup of writes pending in cache cannot occur.

## **2.3. Known Bad Practices**

The following are observed practices that should be avoided as much as is possible as they can lead to performance issues, management problems and customer satisfaction issues.

### **2.3.1. I/O Load concentration**

The tendency to concentrate the I/O load of a server on the smallest possible number of array groups. This can contribute to congestion at the array group level when a server is very active.

### **2.3.2. “Anywhere it can be Found”**

When more space is needed, and all array groups are no longer available, storage is apparently deployed using the “anywhere it can be found” rule, that is, scattered all over the subsystem.

### **2.3.3. Overly dispersed**

Overly diverse emulation; RAID and LDEV layouts: Diversity is appropriate when it is driven by business requirements; however, every effort should be made to reduce excessive diversity. Excessive diversity can make it difficult to manage storage inventory, manage growth, and troubleshoot issues. It is a recommended practice to standardize on the fewest number of combinations required to satisfy specific application requirements.

### **2.3.4. Workload Type Versus RAID Types**

Using an inappropriate workload for the underlying array group, for example. high random writes over 7+1 RAID-5 instead of RAID-10.

### **2.3.5. Uneven Workload Distribution**

Uneven distribution of workload across front-end ports due to hosts being directly connected to the storage system, and thus switches are not used. As long as this remains true, it is unlikely that the distribution of load across front-end ports will ever be even. In the absence of switches, storage administrators should simply aim to keep their front-end ports within the recommended utilization range. As the demand for I/O on the storage system increases, and assuming that the increase in demand comes from adding additional servers, it may at some point become advisable to introduce switches to allow groups of multiple small servers to be consolidated onto single port pairs or quads and thereby improve the balance of port utilization among ports, and thereby increase the overall front end port capacity available for use.

### **2.3.6. Array Groups to Host Segregation**

Segregates host I/O by assigning a small number of separate array groups to each host. This practice prevents any host from being able to impact the I/O of any of the other hosts within the array group layer. While isolation has its benefits, there is an alternative practice that can increase the I/O burst capacity available to each server see Storage Pool discussion.

### 3. Storage Pool Discussion

It is recommended that storage administrators incorporate storage pools, standard emulation types and standard RAID types into their storage management planning. Standard emulation and RAID types will form the basis for storage units of allocation. The units of allocation can then be brought together in a balanced fashion using a Logical Volume Manager. Ultimately these standards will drive recommended practice in activities such as planning, purchasing, deployment, and operations.

It is recommended practice to standardize on a few different combinations that satisfy business requirements. The strategy should focus on:

- Implement Storage Pools
- Standardized emulation types and RAID types
- LVM use and standards

#### **3.1. *Implement Storage Pools***

Storage Pools are a management concept akin to “divide and conquer”. They are not a product feature or anything more than a storage administration discipline. A Storage Pool is a collection of array groups that are managed as a unit and service a defined group of applications with compatible I/O profiles and service objectives.

The Storage Pool approach encourages distributing the I/O of each application across the pool, rather than restricting an application to the minimum number of array groups required to supply the requisite space. It also seeks to limit the dispersal of the activity to the storage pool or a subset of the storage pool. In essence, disperse the I/O, but do not scatter it beyond a justified dispersal.

The administrative objective is to:

- Provide each member of the pool with access to as much array group bandwidth (IOPS and MB/s) as it is likely to require, even in exceptional circumstances
- Distribute I/O as evenly as possible within the pool
- Keep the storage allocations for the group of applications within the pool

It is not necessary to disperse the I/O of every application in the pool across every array group in the pool. In fact it is common to define subsets of the pool as I/O dispersal groups. Nonetheless, it is generally beneficial to distribute the I/O of each application across more than the minimum number of array groups required to provide the storage space. This in turn means that applications typically engage in managed sharing of array group resources.

This approach gives each application access to a larger maximum storage bandwidth capacity from the array groups it uses. This benefit arises from the premise that it is unlikely that every application in a dispersal group will have its peak bandwidth requirements at the same moment.<sup>1</sup>

Storage Pools also share an available space pool, for example, space is added to the pool, one or more array groups at a time. Considering the Logical Volume Manager recommendations earlier in this report, capacity will probably be added to Storage Pools four array groups at a time. This fact alone makes a large number of small pools inappropriate.

Storage Pools should be large enough to achieve I/O dispersal and bandwidth sharing among applications, and large enough to reasonably avoid excessive fragmentation of the available space pool. Storage pools should be small enough to be manageable.

---

<sup>1</sup> *Data networks often incorporate a similar approach in their design. The head end of a frame relay network for a national retail chain (star topology) will generally have ½ to ¼ of the bandwidth of the sum of the store bandwidths because it is unlikely that every store will be active at exactly the same moment.*

Successful Storage Pool implementations require judgment, planning, moderation, and compromise. They involve balancing contending objectives, not the rigid application of rules.

### **3.2. Standardize Emulation Types and Sizes**

Having multiple emulation types and RAID types is appropriate as long as they exist to satisfy business requirements and do not overly complicate the management of the storage system. A recommended practice is to begin by considering an OPEN-V emulation of 36GB<sup>2</sup> for new storage requirements as well as for the re-layout of existing host data. OPEN-V emulation of 36GB is a good choice for a number of reasons.

- 36GB is evenly divisible into Array Group sizes of 288GB (4\*72GB) and 576GB (4\*144GB)
- 36GB is not so small that it would result in a challenging number LDEVs to manage
- 36GB LDEVs can be striped together with an LVM to provide standard database volume sizes<sup>3</sup>
  - For databases between 300GB-900GB in size, a good standard database volume deployment size is 144GB (one 36GB LDEV from an Array Group on each ACP, done twice). This results in roughly two (2) volumes (eight (8) LDEVs) for a 300GB database and 6 volumes (24 LDEVs) for a 900GB database.
  - For databases over 900GB in size, Logical Unit Size Expansions (LUSEs) or adding Host Logical Volume Stripe Columns can be considered to increase the scale of Host Logical Volumes.
    - The use of LUSE has the benefit of simplifying the host environment by reducing the number of devices and volumes that need to be managed within the LVM. LUSE LUNs should be formed from contiguous LDEVs from the same array group.
    - The recommended practice suggests using a Host logical Volume Manager to distribute I/O among array groups via striping rather than LUSE which is a concatenation.
  - If smaller LDEV sizes are required, Concurrent Versions System (CVS) can be used in combination with OPEN-V emulation to create smaller sizes.
- OPEN-V emulation eases the upgrade path to future Hitachi storage hardware such as the Universal Storage Platform

---

<sup>2</sup> Disk drives are like dimensioned lumber in that the nominal size only approximately reflects the actual size. A 2x4 measures 1.5" by 3.5", not 2" by 4". Similarly, the term "36GB" used here is a nominal size intended to reflect ½ of a 72GB disk drive or ¼ of a 144GB drive.

<sup>3</sup> The analysis of customer databases is outside the scope of this document and a complete recommendation with regard to standard database volume sizes for deployment is not possible.

### 3.3. RAID Types

RAID types should be selected based upon the I/O profile of the host applications utilizing the storage. Generally speaking, RAID-5 on a 3+1 Array Group is a good place to start as it is economical and suitable for many applications. RAID-5 is also very good for reads and sequential writes. In cases where an Array Group must write more than 100 tracks per second AND a significant portion of the I/O is random writes, RAID-1 should be considered. RAID-1 handles random writes better than RAID-5. RAID-5 on a 7+1 Array Group is even more sensitive to random writes. 7+1 RAID-5 performs best for applications with a high percentage of sequential operations and a very low percentage of random writes.

### 3.4. Other Considerations

Storage is a shared resource, not unlike network bandwidth. The impact of a new or increasing workload on other workloads should be considered when engineering the storage system or when performing activities that are outside the normal load profile. For example, activities like database loads/refreshes can result in a short term, intense impact on the bandwidth utilization of storage system resources and can thereby create contention for the now scarce resources. Additionally, the aggregate workload generated by multiple hosts commencing activities at the same time can also result in an intense impact on bandwidth utilization.

When contention for scarce resources occurs, it generally impacts other applications sharing that resource. In severe cases of backend write contention that drive the writes pending levels to 70%, contention can impact every application attached to the Lightning 9980V system.

- Balancing the storage system as described above can help to mitigate some of these “out of the norm” activities
- Establishing operational change control procedures can serve to manage these types of events. Change control procedures. This would enable the IT organization to make appropriate business decisions to schedule these activities and/or to ensure that these activities made the most efficient use of available resources
- Staggering the start times of scheduled activities can help to minimize the impact of these activities by spreading the resource demand across a longer timeframe

The storage deployment practice (below) observed at a customer site:

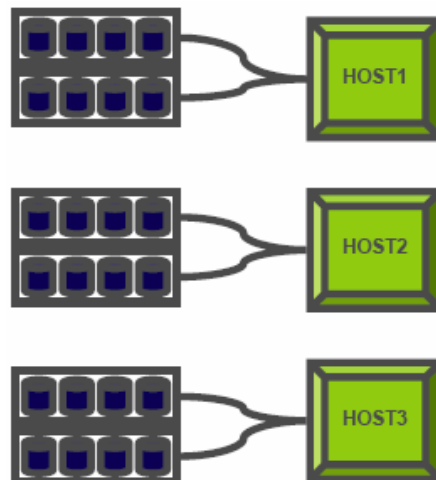
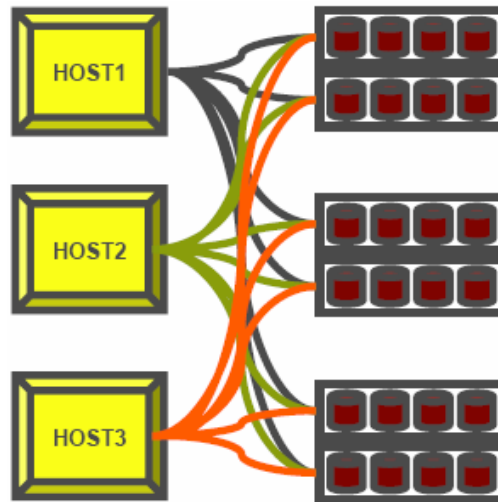


Figure 1 – Example of Storage Deployment by Separate Array Groups

The alternative practice (below) involves interleaving the I/O demand of different, compatible servers across a Storage Pool:



**Figure 2 – Example of Storage Deployment by use of Storage Pools**

Compatible I/O profiles means compatible service objectives, demand cycles, and request sizes.

- Demand cycles are most compatible when different servers sharing the same resources have their peak demands at different times.
- Compatible request sizes means not mixing small block IO with large block IO on the same array group and/or front-end port at the same time, for example more than a difference of several doubles, such as, 8k to 64K
- Compatible service objectives means not having applications that seek minimum response time share the same resources with applications that seek maximum throughput.

The Storage Pool approach encourages distributing the I/O of each application across the pool, rather than restricting an application to the minimum number of array groups required to supply the requisite space.

## 4. Summary

The recommendations provided in this whitepaper should provide storage administrators with storage and application considerations and some opportunities for improving storage performance. Unfortunately there is no one optimal storage configuration. Each customer's storage, SAN, servers, applications and business priority are factors in determining service levels and storage requirements.

In addition to these guidelines, Hitachi Data Systems offers a range of software and services to help manage and maintain your storage environment. GSS is available for a Storage Platform Assessment Service to provide in-depth reporting and analysis of your environment. Software such as Hitachi Tuning Manager software, and associated user training from HDS Academy, may also provide some performance reporting for customers who want to monitor and manage their own storage. Contact your Hitachi Data Systems representative for more information.