HITACHI
DATA SYSTEMS

Hitachi Adaptable Modular Storage AMS2000 Family: Architecture and Concepts

A White Paper

By Alan Benway
Master Performance Consultant

Hitachi Data Systems
Performance Measurement Group,
Enterprise Labs, Technical Operations
Santa Clara, CA 95050

September 17 2008

Notices and Disclaimer

This document has been reviewed for accuracy as of the date of initial publication. Hitachi Data Systems, Inc. may make improvements and/or changes in product and/or programs at any time without notice.

THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESSED OR IMPLIED. Hitachi Data Systems, Inc. EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR PARTICULAR PURPOSE OR NON-INFRINGMENT.

The performance data contained herein was obtained in a controlled, isolated environment. Actual results that may be obtained in other operating environments may vary significantly. While Hitachi Data Systems, Inc. has reviewed each item for accuracy in a specific situation, there is no guarantee that the same of similar results will be obtained elsewhere.

Adaptable Modular Storage® is a registered trademark of Hitachi Data Systems, Inc. in the United States, other countries, or both.

Other company, product or service names may be trademarks or service marks of others.

Table of Contents

I. AMS2000 Family Overview

The Hitachi Adaptable Modular Storage models 2100, 2300, and 2500 (AMS2100, AMS2300, and AMS2500 - or *AMS2000 family*) are the replacements for the previous Hitachi *AMS1000 family* (WMS100, AMS200, AMS500, and AMS1000). The AMS2000 Family models have much higher performance than the AMS1000 Family models, and incorporate several significant design changes. In fact, these changes are so dramatic that a new class of enterprise-like midrange array has been created by Hitachi. As a result of these changes, the AMS2000 Family offers significant differences in how they are configured or accessed by hosts. Nearly every concern that needed to be addressed by a system administrator on the AMS1000 Family has been eliminated.

The AMS2000 Family uses the new Hitachi ***Dynamic Load Balancing Controller***. In particular, the use of separate dynamic load balancing designs on both the front-end ports and the back-end disk links is a significant departure from current midrange architectures. In addition, the Fibre Channel loop back-end of the AMS1000 Family has been replaced by a new, matrixed Serial Attached SCSI (SAS) back-end that allows SAS or SATA-II disks to be freely intermixed in a single enclosure type.

There are many hardware and feature changes present in the AMS2000 Family. The major changes include:

1. Active/Active Symmetric front-end design that allows any port to work with either controller
2. Hardware I/O Load Balancing feature (can be disabled) to maintain a more even distribution of I/O workloads between the two controllers
3. SAS Active Matrix Engine back-end architecture with SAS controllers, more paths, and a dynamic matrix connection from the SAS paths to the individual disks
4. More powerful I/O management processors (Intel Xeon Core-Duo)
5. Single enclosure type for SAS and SATA-II disks (freely intermixed). Adding disk enclosures to AMS2000 Family arrays is far simpler then with the previous generation AMS1000 Family products.
6. Greatly simplified RAID Group configuration and improved SATA write performance.
7. Simplified and extended functionality to extend RAID Group capacity while online and grow and shrink LUN capacity *
8. Improved Shadow Image performance with quick Sync/Resync capability*

   * Future release

The overall effect of these changes has created an entirely new class of storage array that has more in common with enterprise arrays than with midrange arrays. There is now a distinct separation of controller functionality into front-end and back-end I/O engines. The I/O management processors (Xeons) and their companion front-end processors (Tachyons) work together across the two controllers to manage all host I/O operations. The back-end now has separate I/O engines to control a mix of SAS and SATA-II disks. These two subsystems are tied together in the middle with central DCTL chips that act as a combination cache Direct Memory Access (DMA) engine and RAID processor (XOR engine).

Active/Active Symmetric Front-end

The new Active/Active Symmetric front-end design of the AMS2000 Family controllers allows for the access of any of the front-end host ports by either controller. Each of the two controllers (Controller-0 and Controller-1) either has two 4Gbit/s FC ports (AMS2100), four 4Gbit/s FC ports (AMS2300), or eight ports (AMS2500). For example, a host accessing a LUN via port-0B on Controller-0 can actually have the I/O request processed completely by Controller-1 without intervention by the processor in Controller-0. The Active/Active Symmetric front-end design allows the use of operating system native path management and host load balancing rather than separately purchase multipathing software (like HDS Hitachi Data Link Manager – HDLM).

Hardware I/O Load Balancing

A completely new feature for the AMS2000 Family is the ability to use hardware load balancing between the controllers. On the AMS2000 Family, all of the back-end I/O for a LUN is managed by the controller that currently manages that LUN. If there is an on-going imbalance of loads between the controllers, such as one operating at 70% busy and the other at 30% busy, the load balancing mechanism will decide to move management of some of the hard hit LUNs to the non-managing controller. This will shift the back-end workload for those LUNs to the underutilized controller. Note this is independent of which host ports are accessing that LUN – a key observation to make.

SAS Active Matrix engine

The AMS2000 Family uses a new back-end architecture that is very different from the one used by the AMS1000 Family. On the AMS1000 Family (see Appendix A-D for architecture overviews), the DCTL RAID chip was directly connected to the back-end Tachyon DX2 (2Gbit/s FC-AL) interface chips. On the AMS2000 Family controller board, the enhanced DCTL chip sends commands to a powerful companion *SAS I/O Controller* processor (IOC).

Processor changes

The AMS 1000 Family uses a PowerPC 7447a processor and chipset. All paths within the controller were PCI-X, a 533MB/s protocol (wire speed). The AMS2000 Family has moved up to an Intel Xeon (core duo) CPU and chipset that uses PCI-express (PCI-e) 8-lane busses operating at an aggregate of 2000 MB/s (wire speed). Though the PowerPC and Xeon chips have similar clock speeds, the Xeon Core Duo design has far more power due to the much faster system bus (667MHz vs. 166MHz, controlling access to local RAM and the Intel *MCH* Memory and I/O Controller chip), single or dual CPU cores per chip, and a much higher degree of execution parallelism within each core.

Enclosure changes

Another major change is that the AMS2000 Family has a single type of enclosure common to both SAS and SATA disks. Previously on the AMS1000 Family, there were separate enclosure types needed for FC and SATA disks. In the AMS2000 Family, both disk types may be intermixed in the same enclosure. The two *expander units* in each enclosure for the AMS2100 and AMS 2300 are part of the new back-end disk matrix system. Also, the AMS2000 Family enclosure no longer has address switches – something that could cause installation headaches on the AMS 1000 Family when not properly set. As a last note, in the AMS1000 Family, care had to be taken

to put SATA HDDs in certain slots within the tray (for dispersed disk selection), and this requirement is completely eliminated in the AMS2000 Family.

RAID Group layouts

The new back-end architecture has eliminated all of the issues relating to the manual assignment of disks to RAID Groups on the AMS 1000 Family. Previously, one had to pay a lot of attention to the specific mix of disks and enclosures selected when configuring a new RAID Group. Further complicating the issue, the layout choices were different when using FC or SATA disks.

II. Architecture Overviews by Model

This section will discuss the general architecture overviews of each model. Expanded discussion of the internal elements is in the next section. The AMS2100 replaces the AMS500, and the AMS2300 replaces the AMS1000. The AMS2500 is a new high-end model. The table below lists the major differences among the AMS2000 Family. All numbers are for the complete system

Model	Maximum Disks	Maximum Cache (GB)	Maximum FC paths	Disk links	Bandwidth to Cache (overall)
AMS2100	120	4/8	4	16	8 GB/s
AMS2300	240	8/16	8	16	16 GB/s
AMS2500	480	16/32	16	32	16 GB/s

Figure 1. Comparison of AMS2000 models

AMS2100

At the core of the AMS2100 are the eighth generation Hitachi DCTL-S controller (a high performance RAID and I/O engine) and an Intel Xeon processor. The Intel Xeon processor is a significant upgrade from the 32-bit PowerPC processor (PPC 7447A) used in the AMS1000 Family. An AMS2100 consists of two controllers, 4GB or 8GB of cache, and four 4Gbit/s Fibre Channel (FC) host ports controlled by two high performance 2-port DE4 Tachyon processors.

There are 16 3Gbit/s back-end disk paths controlled by the two SAS engines and up to 120 SAS or SATA disks in the system. The disks may be any mixture of 3Gbit/s SAS disks or 3Gbit/s SATA-II disks that the AMS2000 Family supports. The AMS2100 chassis has a built-in 15-disk enclosure (RK) with two RKS controller boards, and the system can grow by up to seven external enclosures (RKAK). The first four internal disks installed in each system are for use by the microcode in managing the configuration images, and must be of the same type (SATA-II or SAS). Each external enclosure (RKAK) must have at least two disks installed.

The connection bandwidth between each DCTL-S chip and its cache is 4GB/sec. The AMS2100 has one bank of cache (1 DIMM) per controller, so total cache will either be 4GB (2GB DIMMs) or 8GB (4GB DIMMs). There is a 2GB/sec PCIe communications bus between the two DCTL-S processors for inter-controller communications, maintenance operations, and duplexed (mirrored) cache write operations.

Each AMS2100 controller includes:

- A DCTL-S processor (the I/O "pump" with RAID XOR functions)
- A 1.67GHz Intel "Sossaman" Dual-Core Xeon LV series (low voltage) processor (single core version) and 1GB of memory. This processor is the microcode engine, or the I/O management "brains"
- 4GB (2GB DIMMs) or 8GB (4GB DIMMs) of cache per system
- Two high-performance Tachyon DE4 2-port 4Gbit/s Fibre Channel processors controlling the front-end host connections
- Two SAS controllers servicing the sixteen active back-end SAS disk links
- All internal busses are now 2048 MB/sec 8-lane PCI Express (PCIe) instead of the previous 533 MB/sec PCI-x bus

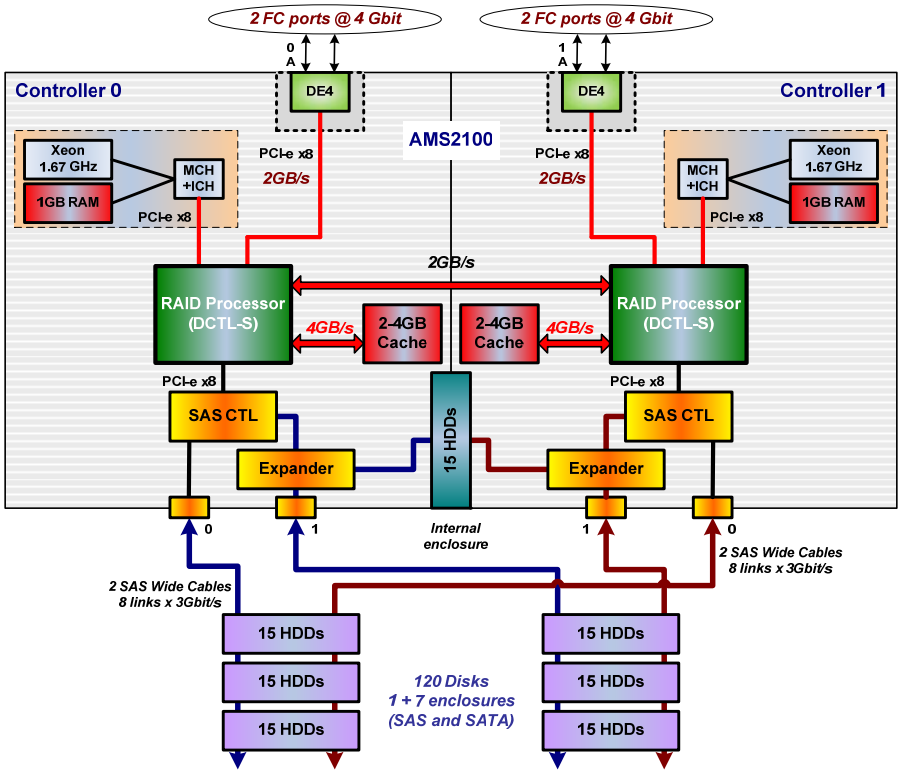


Figure 2. AMS2100 overview

AMS2300

At the core of the AMS2300 are the eighth generation Hitachi DCTL-H controller (a high performance RAID and I/O engine) and an Intel Xeon processor. The Intel Xeon processor is a significant upgrade from the 32-bit PowerPC processor (PPC 7447A) used in the AMS1000 Family. The AMS2300 consists of two controllers, 8GB or 16GB of cache, and eight 4Gbit/s host ports controlled by two high performance 4-port QE4 Tachyon processors.

There are 16 3Gbit/s back-end disk paths controlled by the two SAS engines and up to 240 SAS or SATA disks in the system. The disks may be any mixture of 3Gbit/s SAS disks or 3Gbit/s SATA-II disks that the AMS2000 Family supports. The AMS2100 chassis has a built-in 15-disk enclosure with two RKM controller boards, and the system can grow by up to fifteen external enclosures.

The first four internal disks installed in each system are for use by the microcode in managing the configuration images, and must be of the same type (SATA-II or SAS). Each external enclosure must have at least two disks installed.

The connection bandwidth between each DCTL-H processor and cache is 8GB/sec. The AMS2300 has two banks of cache (1 DIMM each) per controller, so total cache will either be 8GB (2GB DIMMs) or 16GB (4GB DIMMs). There is a 2GB/sec PCIe communications bus between the two DCTL-H processors for inter-controller communications, maintenance operations, and duplexed (mirrored) cache write operations.

Each AMS2300 controller includes:

- A DCTL-H processor (the I/O "pump" with RAID XOR functions)
- A 1.67GHz Intel "Sossaman" Dual-Core Xeon LV series (low voltage) processor (single core version) and 1GB of local memory. This processor is the microcode engine, or the I/O management "brains".
- 8GB (2GB DIMMs) or 16GB (4GB DIMMs) of cache per system
- Two high performance Tachyon QE4 4-port 4Gbit Fibre Channel processors controlling the 8 front-end host connections
- Two SAS controllers servicing the sixteen active back-end SAS disk links
- All internal busses are now 2048 MB/sec 8-lane PCI Express (PCIe) instead of the previous 533 MB/sec PCI-x bus

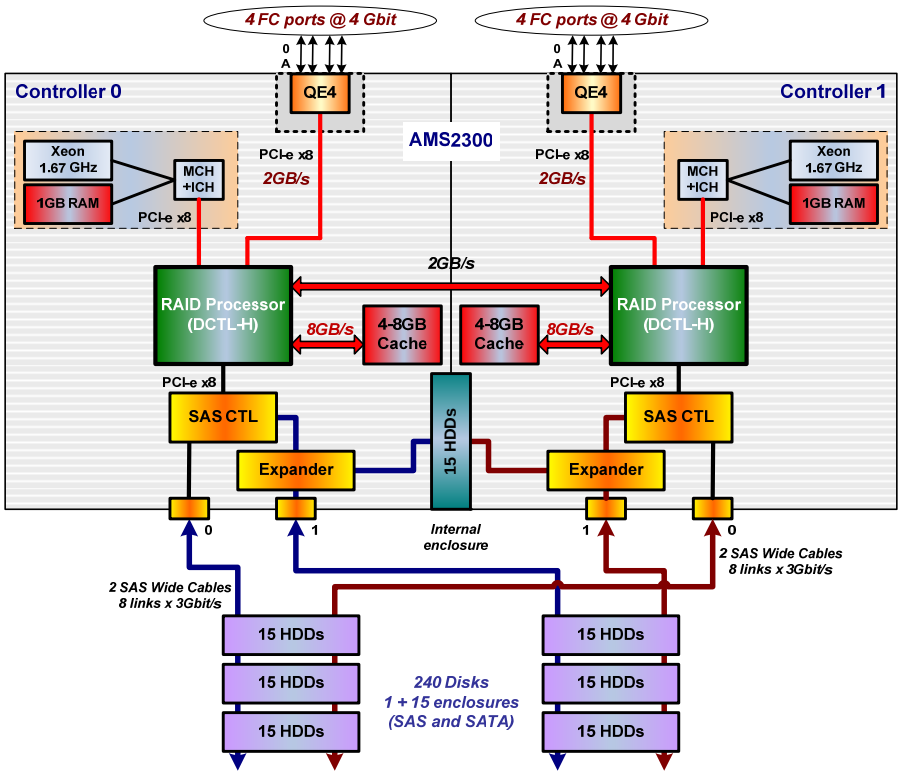


Figure 3. AMS2300 overview

The AMS2300 looks very similar to the AMS2100. The actual differences (2100 vs. 2300) are as follows:

- The AMS2100 uses DCTL-S (RAID and I/O processor) used instead of the DCTL-H used on the AMS2300.
- The AMS2100 uses 2 banks of cache with one DIMM slot each (8GB) vs. 4 banks of cache used on the AMS2300. This is with optional cache installed with one DIMM slot each (16GB)
- The AMS2100 uses the 2-port DE4 Tachyon processor (4 ports) instead of the 4-port QE4 Tachyon processor (8 ports) used on the AMS2300.
- The AMS2100 has 120 disks vs. 240 disks on the AMS2300.

AMS2500

At the core of the AMS2500 are the eighth generation Hitachi DCTL-H controller (a high performance RAID and I/O engine) and an Intel Xeon dual core processor. The Intel Xeon dual core processor is a significant upgrade from the 32-bit PowerPC processor (PPC 7447A) used in the AMS1000 Family. The AMS2500 consists of two controllers, 16GB or 32GB of cache, and sixteen 4Gbit/s Fibre Channel (FC) host ports controlled by four high performance 4-port QE4 Tachyon processors.

There are 32 back-end 3Gbit/s disk paths controlled by the four SAS engines and up to 480 SAS or SATA disks in 32 disk enclosures (RKAK) of 15 disks each. These enclosures accept either 3Gbit/s SAS disks or 3Gbit/s SATA-II disks. There are no disks in the controller module (RK) on this model. The first RKAK enclosure must have at least four disks of the same type, while any additional RKAK enclosures must have at least two disks each.

The connection bandwidth between each DCTL-H chip and its cache is 8GB/sec. The AMS2500 has two banks of cache (2 DIMMs each) per controller, so total cache will either be 16GB (2GB DIMMs) or 32GB (4GB DIMMs). There is a 2GB/sec PCIe communications buss between the DCTL-H processors for inter-controller communications, maintenance operations, and duplexed (mirrored) cache write operations.

Each AMS2500 controller includes:

- A DCTL-H processor (the I/O “pump” with RAID XOR functions)
- A 2GHz Intel “Sossaman” Dual-Core Xeon LV series (low voltage) processor and 2GB of local memory. This processor is the microcode engine, or the I/O management “brains”.
- 16GB (2GB DIMMs) or 32GB (4GB DIMMs) of cache per system
- Four high performance Tachyon QE4 4-port 4Gbit/s Fibre Channel processors controlling the 16 front-end host connections
- Four SAS controllers servicing the 32 active back-end SAS disk links
- All internal busses are now 2048 MB/sec 8-lane PCI Express (PCIe) instead of the previous 533 MB/sec PCI-x bus

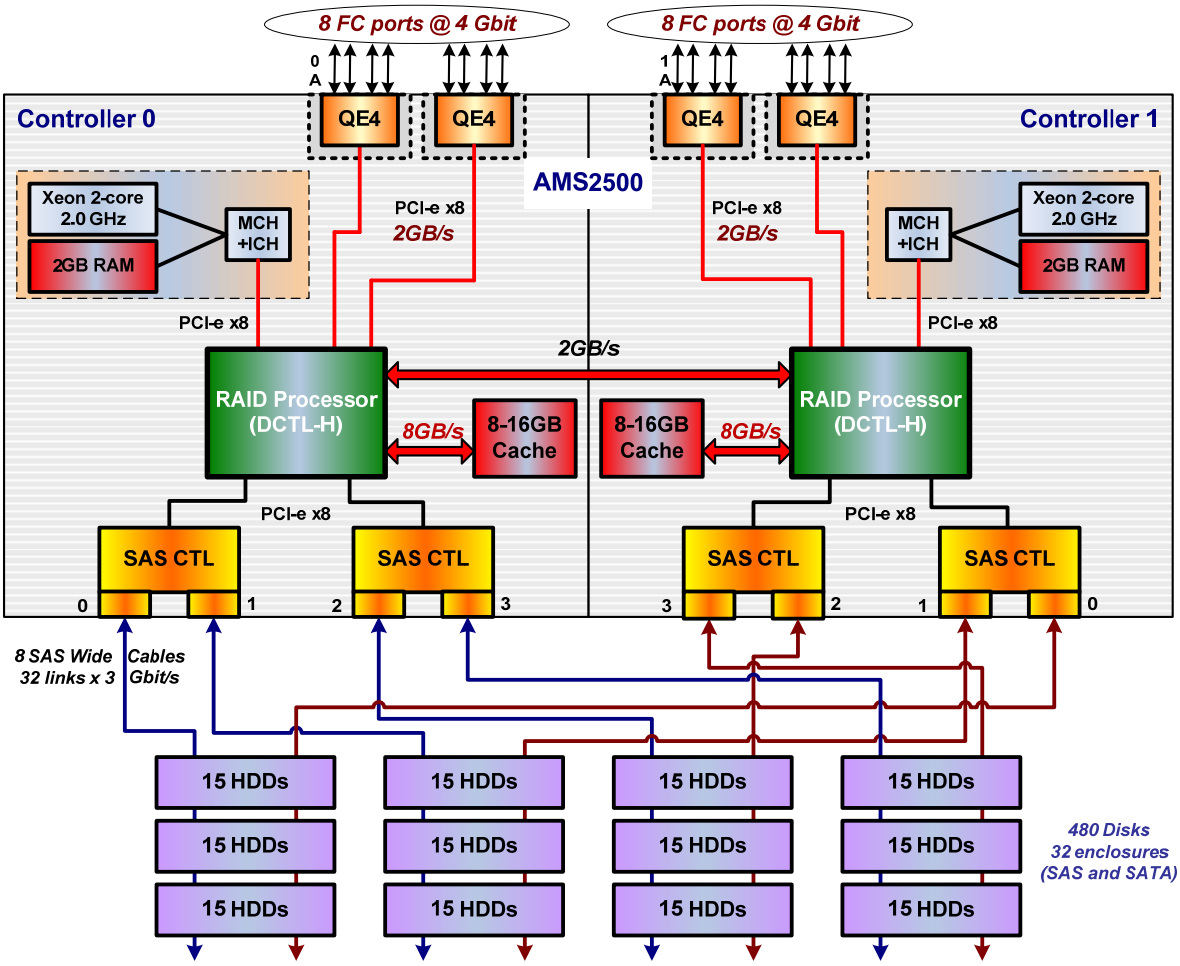


Figure 4. AMS2500 overview

The AMS2500 is a significant upgrade from the AMS2300. The actual differences (2300 vs. 2500) are as follows:

- The AMS2300 uses 1.67GHz single core Xeons and 1GB of RAM vs. 2GHz dual core Xeons with 2GB of RAM on the AMS2500.
- The AMS2300 has 4 banks of cache with one DIMM slot each (16GB max) vs. 4 banks of cache with two DIMM slots each (32GB max) on the AMS2500.
- The AMS2300 has one 4-port QE4 Tachyon processor per controller (8 host ports total) vs. dual 4-port QE4 Tachyon processors (16 host ports) per controller on the AMS2500.
- The AMS2300 has two SAS I/O engines (16 disk links) per system vs. four SAS I/O engines (32 SAS links) per system on the AMS2500.
- 240 disks vs. 480 disks

Comparisons of AMS Features and Limits

The following tables summarize various aspects of the AMS2000 Family, along with matching tables for the AMS1000 Family for comparison.

Configuration Limits

Configuration Limits			AMS 2100	AMS 2300	AMS 2500
RAID Groups			50	75	100
Max LUNs			2048	4096	4096
Port IO request Limit			512	512	512
LUN queue depth					
* SAS			32	32	32
* SATA-II			16	16	16
Max LUN size (TB)			60	60	60
Max spare disks			15	30	30
Cache partitions			16	32	32
Host WWNs per port			128	128	128

Configuration Limits	WMS 100	AMS 200	AMS 500	AMS 1000	
RAID Groups	25	25	45	90	
Max LUNs	512	512	2048	4096	
Port IO request Limit	512	512	512	512	
LUN queue depth					
* FC	na	32	32	32	
* SATA-II	1	1	1	1	
Max LUN size (TB)	2	2	2	2	
Max spare disks	15	15	15	30	
Cache partitions	6	8	16	32	
Host WWNs per port	128	128	128	128	

Controller Details

Controllers			AMS 2100	AMS 2300	AMS 2500
Xeon CPU (per CTRL)			1.67GHz 1-core	1.67GHz 1-core	2GHz 2-core
CPU RAM (per CTRL)			1GB	1GB	2GB
Total Cache (GB)			4 or 8	8 or 16	16 or 32
RAID ASIC			DCTL-S	DCTL-H	DCTL-H
Bus type			PCIe	PCIe	PCIe
Disks (SAS and SATA)			120	240	480
3Gb SAS links			16	16	32
4Gb FC Ports			4	8	16

Note: The cache size is determined by the use of either 2GB or 4GB DIMMs in the controllers. Only one type may be used, so there are only two cache sizes available per array.

Controllers	WMS 100	AMS 200	AMS 500	AMS 1000	
PowerPC CPU	500MHz	500MHz	1GHz	1.7GHz	
CPU RAM	256MB	256MB	512MB	1GB	
Total Cache (GB)	1-2	2-4	2-8	4-16	
RAID ASIC	DCTL-M	DCTL-M	DCTL-M	DCTL-H	
Bus type	PCIx	PCIx	PCIx	PCIx	
Disks (FC max)	n/a	105	225	450	
Disks (SATA-II max)	105	90	210	420	
2Gb FC loops	2	2	4	8	
4Gb FC Ports	2 or 4	2 or 4	2 or 4	4 or 8	
1GE iSCSI Ports	2 or 4	2 or 4	2 or 4	4	

Back-end Details

Back-end			AMS 2100	AMS 2300	AMS 2500
Disk enclosures			8	16	32
* internal			1	1	na
* external			7	15	32
Disks (SAS, SATA-II)			120	240	480
* internal			4-15	4-15	na
* external			2-105	2-225	4-480
3Gb SAS links			16	16	32
Avg disks per SAS link			15	30	30

Back-end	WMS 100	AMS 200	AMS 500	AMS 1000	
Disk enclosures	7	7	15	30	
* internal	1	1	1	na	
* external	6	6	14	30	
Disks (FC or SATA-II)	105 SATA-II	105	225	450	
* internal	5-15	5-15 FC	5-15 FC	na	
* external	2-90	2-90	2-210	5-450**	
2Gb FC loops	2*	2*	4	8	
Avg disks per loop pair	52	52	112	112	

* = there are four back-end FC loop connectors, but only two loops using FC-AL hubs in front of each pair of paths

** = must have 5 minimum FC HDDs for first tray connected to RKH/HE, 2 minimum for each additional RKA/RKAJAT

Disk and RAID level Details

The table below shows the RAID levels and their minimum-maximum configurations, as well as all disks types available.

RAID Levels	minimum	maximum
RAID-1	1D+1D	-
RAID-10	2D+2D	8D+8D
RAID-5	3D+1P	15D+1P
RAID-6	2D+2P	28D+2P

DISKS	Type	Model
	SAS	146GB 15k
	SAS	300GB 15k
	SAS	400GB 10k
	SATA-II	1TB 7.2k

As mentioned above, the DCTL processor is the I/O “pump” for each controller. It works in conjunction with the Xeon processor CPU, which runs the microcode and makes all determinations about I/O processing. The DCTL is a RAID XOR (parity) processor that creates all parity for RAID-5 or RAID-6 writes. It is also the DMA path from the front-end components (Xeon CPU, Tachyon chips) to the data cache. The two DCTL processors in a system have a private 2GB/s communications bus (PCIe 8 lane) over which they communicate status and pass the mirrored write blocks.

II. Architecture Concepts and Details

This section discusses some of the details or usage options of the new design. This includes:

- Tachyon processor features
- Active/Active Symmetric front-end design
- Hardware I/O Load Balancing feature
- SAS Active Matrix Engine
- Simple RAID Group configuration

Tachyon Processor Overview

A Tachyon processor (single chip) is used to bridge the fibre channel host connection to a usable form for internal use by the Xeon and DCTL processors. Various types of Tachyon processors are used as the fibre channel controllers on server HBAs and storage arrays. Tachyon processors are used for one of two fundamental purposes:

- Drive or terminate the fibre channel protocol transport layer used between a server port and a storage port
- Drive or terminate the fibre channel protocol transport layer used between a disk controller and an FC disk on an Arbitrated-Loop (FC-AL).
- Perform certain housekeeping functions on the FC packets

The AMS2000 Family uses either the DE4 Tachyon processor (AMS2100) or the QE4 Tachyon processor (AMS2300, AMS2500). This high-power Tachyon xE4 series processor provides a variety of functions, including:

- A conversion of the FC transport protocol to the PCIe bus for use by one to four controller processors
    - SCSI initiator and target mode support
    - Complete FC protocol sequence segmentation or reassembly
    - Conversion to the PCIe bus protocol
- Provides simultaneous full duplex operations of each of two or four ports
- Multiple DMA transfer modes available for directly writing blocks to cache
- FC Frame Steering (Virtualization support)
- Error detection and reporting
- Packet CRC encode/decode offload engine

The DE4 and QE4 processors can provide very high levels of performance as they are connected to the same high performance, 2GB/sec 8-lane PCI Express (PCIe) bus as the Xeon controller and the DCTL RAID processor. This is four times more bandwidth than provided by the 533 MB/sec PCI-x bus used in the AMS1000 Family. The DE4 and QE4 processors each have far more processing power than the Tachyon DX4 processor that was used in the AMS1000 Family.

The DE4 processor can drive both of its 4Gbit/s ports at full speed, whereas the older DX4 chip can only drive one of its two 4Gbit/s ports at full speed. The QE4 processor can drive all four of its 4Gbit/s ports at full speed. On the AMS1000 Family, one had to be careful about mapping LUNs to ports and which ports to use on the attached servers in order to balance the loads evenly across the DX4 chips.

Here is a generalized view of the 4-port QE4 processor from PMC-Sierra literature (the DE4 is the same layout but with only two ports):

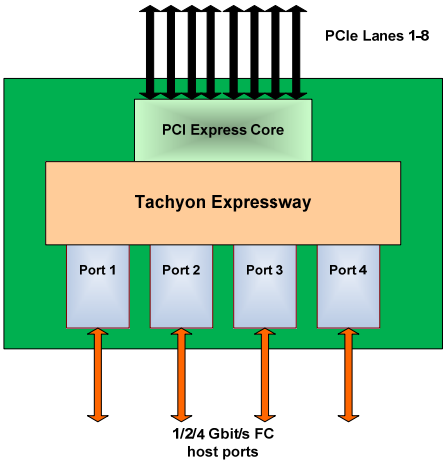


Figure 5. QE4 Tachyon processor overview

Active/Active Symmetric Front-end Design

The AMS2000 Family introduces ***Active/Active Symmetric*** **front-end design**, a totally new concept to the midrange array arena. The rigid concept of ***LUN ownership*** by controller has been replaced with a more powerful method of ***LUN Management***. Rather than a simple LUN ownership by controller, now there is a dynamic, global table of all configured LUNS that determines which controller will execute an I/O request for a LUN. This control list is independent of which front-end port (either controller) is involved in the host I/O request. The Active/Active Symmetric front-end enables this new capability and the corresponding freedom from micro-managing the appearance of LUNs on certain paths for certain hosts.

All LUNs are initially automatically assigned by Storage Navigator Modular 2 on a round-robin basis to the controller I/O management lists as LUNs are created. The table of I/O management is changed over time by the operation of the Hardware I/O Load Balancing feature (described below) that, if enabled, will remap certain LUNS from one controller's management list to the other controller.

The ***Active/Active Symmetric*** controller front-end architecture allows for the access of any of the front-end host ports on either controller by either Xeon processor. This design puts the Xeon processors, the DE4/QE4 Tachyon processors, and the DCTL chip into a partnership that manages all host I/Os. The Tachyon processors directly send each incoming I/O request (one or more fibre channel packets) to the proper Xeon processor from either controller. Likewise, the Xeon processors know which Tachyon to send a response to for each host I/O request. The Xeon processors manage all I/O operations within a controller for the set of LUNs on their management list. The DCTL chip works with both the Xeon and Tachyon processors to facilitate the DMA data transfers (controlled by the Tachyon processors) in and out of cache, as well as to perform the RAID XOR functions and write duplexing to mirrored cache regions. The Tachyon processors have lists for all LUNs in the array so that they know which Xeon processor is currently responsible for managing a LUN.

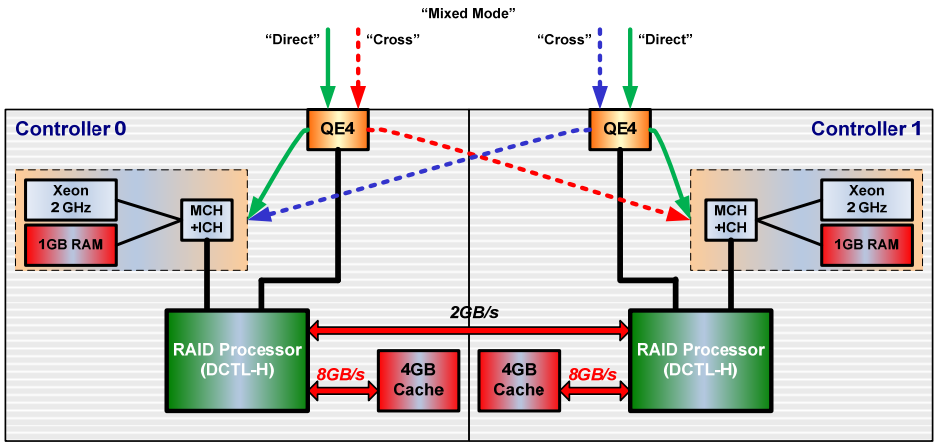


Figure 6. Active/Active Symmetric feature

For example (see the red dotted line in Figure 6), a host accessing a LUN via port-0B on Controller-0 can actually have the I/O request processed completely by Controller-1 without intervention by Controller-0's Xeon processor DCTL chip, or cache (except for mirrored writes). When Controller-1 is finished, the response is routed directly back from the Xeon processor across its PCIe bus to port-0B on the Tachyon processor.

On the AMS1000 Family the controller front-end design was ***Active/Active Asymmetric.*** Host I/O requests are passed across to the owning controller through the DCTL chip interconnect bus for execution and then handed back the same way when the I/O is complete (*Data Share* mode – see Appendix E for further discussion).

Figure 6 above illustrates the two types of Active/Active Symmetric operations that occur on the AMS2000 Family. When an inbound I/O request arrives at a port on the same controller that currently manages that LUN, the I/O operation is called "**Direct Mode**" (shown by the green traces). When an inbound I/O request arrives at a port on the controller that does *not* manage that LUN, this is referred to as "**Cross Mode**". The red trace shows an I/O from a host on a port on Controller-0 bound for a LUN currently managed by Controller-1. The blue trace shows an I/O on Controller-1 that needs to be executed by Controller-0 since it currently manages that LUN. When both "Cross" and "Direct" I/O modes are present, this is called "**Mixed Mode**". In general, there was a fairly small overhead measured (1-4% typically) for a Cross mode test when running a 50:50 mix of Direct and Cross, using a large number of LUNs, and running heavy test workloads. Note that because of the packet steering capability of the Tachyon processor, there is no traffic across the inter-DCTL bus for processing Cross Mode I/Os (except for mirrored write blocks). The Tachyon uses the shared PCIe bus in front of the DCTL chip.

Host I/O Load Balancing

Since the design of the AMS2000 Family is Active/Active Symmetric, operating system native path management, such as Microsoft MPIO, Solaris MPIO, AIX MPxIO for example, including the various load balancing algorithms, are fully supported. In addition, Veritas DMP and Hitachi's HDLM are also fully supported. There is no need for host path assignment management between the ports on the two controllers.

Hardware I/O Load Balancing (HLB) Feature

Hardware Load Balancing (**HLB**) is a new AMS2000 feature that is distinctly different from the Active/Active Symmetric mode, but it makes use of those capabilities. HLB is the automatic change to the controller management tables of one or more LUNs due to a sustained imbalance of CPU busy rates between the two controllers. Note that this does not affect the mapping of LUNs to front-end ports (they remain where they are), only the assignment of which controller processes those I/O requests. After there has been a significant controller CPU "delta percent" busy imbalance (trigger 1) between the controllers for some period of time (trigger 2), the HLB mechanism will activate and decide which LUNs are the best ones to move to the other controller's I/O management list in order to bring about a balance of controller CPU loads.

The next time the HLB mechanism decides to rebalance, any LUNs are candidates for an I/O management change. If a LUN is being accessed by a front-end port located on the non-managing controller, then these I/O requests are directly cross-mapped (via the Active/Active Symmetric mechanism) from the local QE4 Tachyon processor across to the Xeon CPU in the managing controller. This is because each controller must process all I/O requests for those LUNs it currently manages. There is no more *Data Share* hand-off mode (see Appendix E) between the DCTL processors as was the case on the AMS1000 Family. Based on current test results, it appears that the overhead cost of this Cross-mode is in the 1-4% range.

Each DE4/QE4 processor can interact (over its 2GB/s PCIe bus interface) with up to four separate CPUs across the two controllers. There is one CPU per controller in the AMS2100 and AMS2300 (single core Xeons), and two CPUs per controller in the AMS2500 controller (single processor, dual core Xeons). This, plus the ability of a DE4/QE4 to steer FC packets to any attached processor is part of the Load Balancing system. Note that each DE4/QE4 operates as a companion processor to the Xeons over their common PCIe bus, rather than a directly connected slave chip. A useful application of this feature is when a management processor (the Xeon CPU) is rebooting for a microcode upgrade or certain configuration changes, this action does not also reset the DE4/QE4 processor(s) on that same controller. However, the Tachyon processor(s) is signaled to send all I/O to the other controller while the reboot is in progress. All of the I/O that would have been processed by that rebooting controller (due to the global LUN management tables) is temporarily handed off to the other controller.

SAS Active Matrix Engine

The AMS2000 Family uses a new back-end architecture that is very different from the one used by the AMS1000 Family. On the AMS1000 Family (see Appendix A-D for architecture overviews), the DCTL RAID chip was directly connected to the back-end Tachyon DX2 (2Gbit/s FC-AL) interface chips. On the AMS2000 Family controller board, the enhanced DCTL chip sends commands to a powerful companion ***SAS I/O Controller*** **processor (IOC)**. Each IOC processor contains dual CPUs, the ability to directly drive SAS or SATA-II disks, and provides (8) 3Gbit/s SAS links. In a major departure from standard FC-AL back-end designs, the SAS IOC processor will dynamically select one of the 8 SAS links over which to direct an I/O request to a disk. The SAS IOC maintains an even balance of loads across these 8 back-end paths.

Downstream of each IOC processor is a 24-port ***SAS Expander Processor*** **(SXP)**. The SXP decides which of the four 3Gbit/s SAS links (paths) from the SAS IOC to cross-connect to an individual disk for an I/O operation. This matrix connection feature implements dynamic balancing of the

loads on the four SAS link per IOC. The SXP either directly accesses one of the 15 internal disks in the command module (AMS2100/2300 only), or a disk in an external enclosure via the four external SAS link ports. These four ports are controlled by a ***SAS Multiplexor (SMX) interface* chip** that is connected to either the IOC processor or the SXP chip (not on AMS2500). The SMX chip is where each Wide SAS cable is attached to the rear panel of a controller.

The AMS2000 Family has many more back-end paths than the AMS1000 Family. These paths are also 50% faster than the previous models (3Gbit/s instead of 2Gbit/s). Table 1 shows a comparison of the major back-end details of these two generations of midrange systems.

Back-end			AMS 2100	AMS 2300	AMS 2500
Disk enclosures			8	16	32
* internal			1	1	na
* external			7	15	32
Disks (SAS, SATA-II)			120	240	480
* internal			4-15	4-15	na
* external			2-105	2-225	4-480
3Gb SAS links			16	16	32
Avg disks per SAS link			8	15	15

Back-end	WMS 100	AMS 200	AMS 500	AMS 1000	
Disk enclosures	7	7	15	30	
* internal	1	1	1	na	
* external	6	6	14	30	
Disks (FC or SATA-II)	105 SATA-II	105	225	450	
* internal	5-15	5-15 FC	5-15 FC	na	
* external	2-90	2-90	2-210	5-450**	
2Gb FC loops	2*	2*	4	8	
Avg disks per loop pair	52	52	112	112	

Table 1. Comparison of back-end features by models

* = there are four back-end FC loop connectors, but only two loops using FC-AL hubs in front of each pair of paths

** = must have 5 minimum FC HDDs for first tray connected to RKH/HE, 2 minimum for each additional RKA/RKAJAT

SAS Links, Wide Cables, and Expanders

The SAS back-end system is quite different from the previously used multi-loop FC-AL design. There is now the concept of a **SAS wide cable** which contains 4 separate **3Gbit/s SAS links** within it. Each controller has either two (AMS2100/2300) or four (AMS2500) SAS Wide connectors. Each external disk enclosure has four connectors for these SAS Wide cables, with two on the controller side and two on the back for daisy-chaining additional enclosures (see Figure 7 below).

The AMS2100/2300 controllers and the external disk enclosures all have “SAS Expander” chips that create a matrix system enabling a SAS IOC processor controller to access any disk in an enclosure over any of the four SAS links in a SAS cable. This is done dynamically under the control of the SAS controller chip in the controller. This means that the controller will determine which actual link to use when connecting to a disk in order to balance out the loads on the back-end. In comparison, on the AMS1000 Family, the back-end FC-AL path used by a controller was fixed to disk slots in the enclosure. There was no equivalent chip in the AMS1000 Family for FC-

AL to the SAS controller chip. The DX2 Tachyon processor was directly connected to the DCTL chip.

Inside each disk enclosure there is a pair of "**expander**" units. These may be viewed as two 4x15 switched matrixes attached to the two SAS wide cables per enclosure that cross-connect the disks to all eight of the SAS links. There are 4 ports entering the expander, 4 ports passing through to the next expander in another enclosure, and 15 ports per expander to which one side of each disk canister is connected. The expander determines which of its four SAS links (the SAS wide cable) is actually used to talk to a particular disk each time. Note that each AMS2100/2300 controller has one expander chip built in. These chips control access from the SAS IOC to the 15 internal disks (there is no internal enclosure, just disk slots). Each expander has a SAS wide cable connection coming in the front side and passing through the back side. Enclosures are daisy-chained together using these connectors. The same is true for the internal expanders – they have an 'outbound' SMX connector to attach to external enclosures as well.

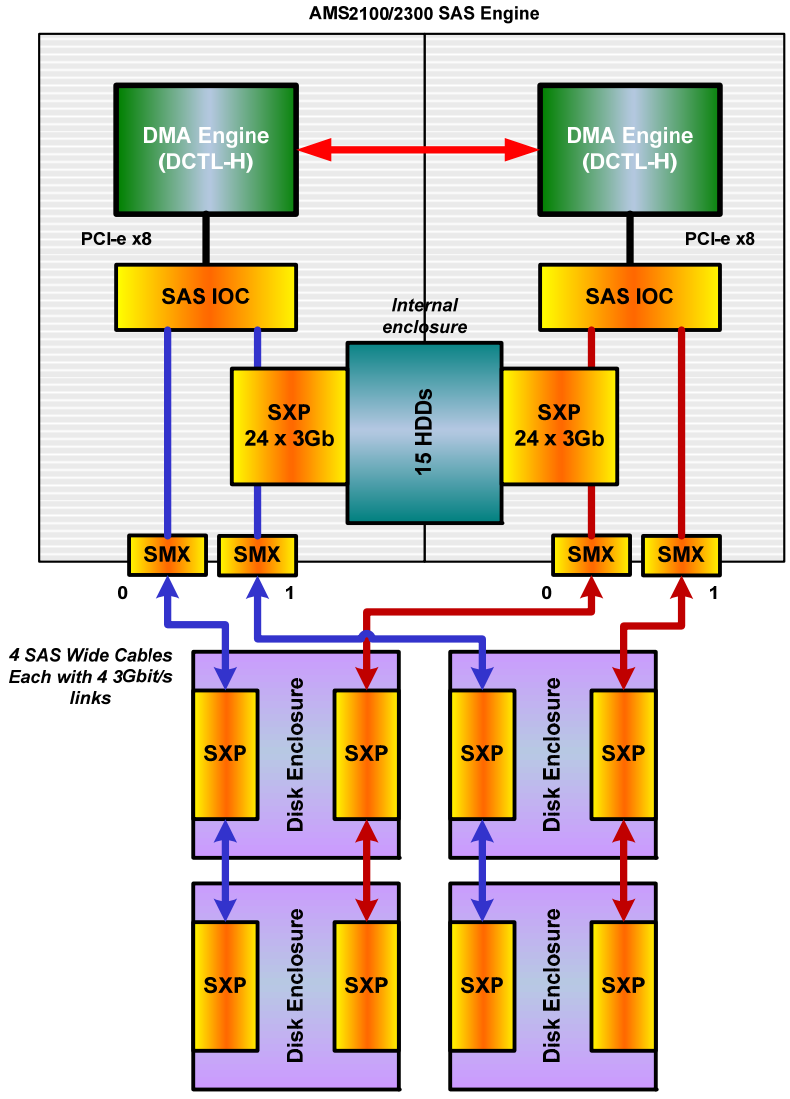


Figure 7. SAS disk matrix system

SAS/SATA-II External Enclosure

Another major change is that there is now a single type of enclosure common to both SAS and SATA disks. Previously in the AMS1000 series, there were different enclosure types needed for FC and SATA disks. Figure 8 depicts a simplified view of the disk enclosure details.

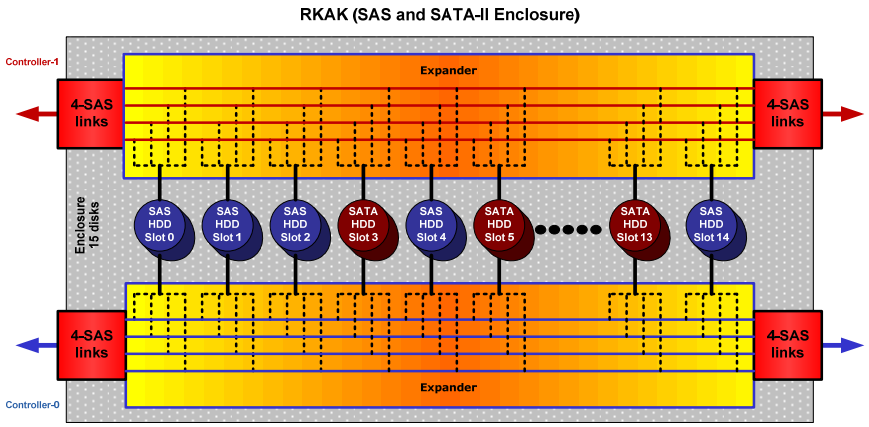


Figure 8. SAS/SATA disk enclosure

Adding disk enclosures to AMS2000 Family arrays is far simpler then with the previous generation AMS1000 Family products. When adding enclosures to an AMS1000 Family, one used pairs of SATA enclosures and single FC enclosures to attach as a "column" to sets of four FC loops. The SATA enclosures had two FC-AL loop attachments while the FC enclosures had four FC-AL loop connection points. Also, each external enclosure had loop ID switches that had to be set correctly in order for the controller to recognize them. There are no such switches on the new enclosure, thus eliminating a possible source of error when configuring a system. Figure 9 illustrates the connection of FC and SATA enclosures to an AMS500. Figure 10 is an overview of the AMS2100 SAS paths and enclosures.

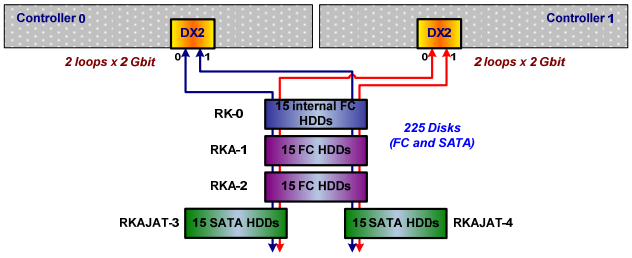


Figure 9. AMS500 back-end loops and enclosures

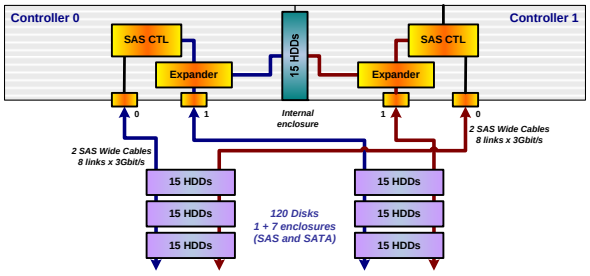


Figure 10. AMS2100 back-end loops and enclosures

Figure 11 illustrates how an AMS1000 is attached to its FC or SATA enclosures, while Figure 12 is for the AMS2300 and Figure 13 is for the AMS2500, both with their single enclosure type. Notice how the AMS2500 has no internal disks.

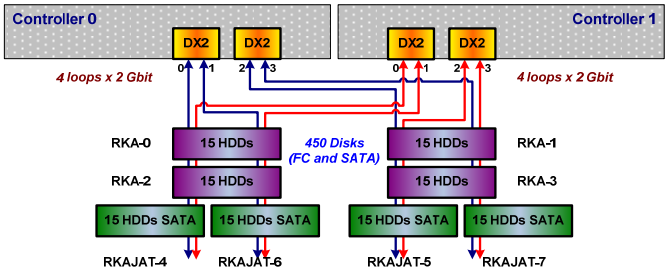


Figure 11. AMS1000 back-end loops and enclosures

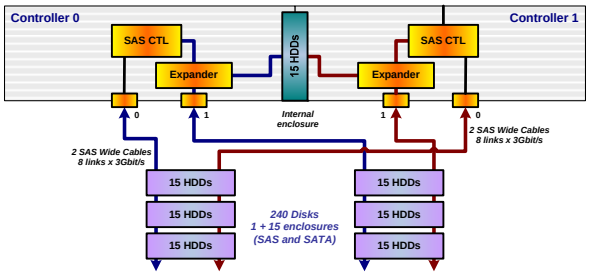


Figure 12. AMS2300 back-end loops and enclosures

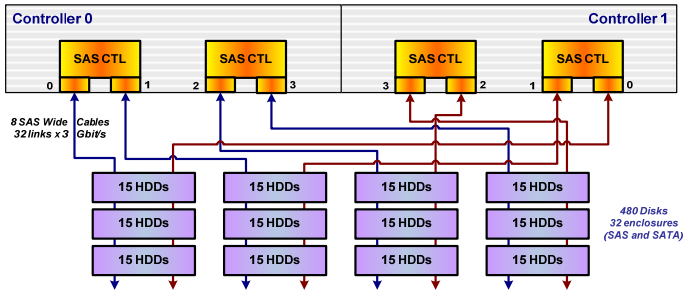


Figure 13. AMS2500 back-end loops and enclosures

Simple RAID Group Configuration

AMS1000 Family RAID group and physical disk configuration was far more complex than the same processes on AMS2000. On the AMS 1000 products, due to the dedicated mapping between back-end FC loops and one of the 2 or 4 internal FC paths in each enclosure, it was necessary to be aware of this direct relationship when assigning disks and laying out RAID Groups. Furthermore, due to the differences between FC and SATA enclosures, and the method of accessing the disks by RAID Group, there was a different layout process required for these two disks types.

On the AMS2000 Family, the assignment of disks to RAID Groups is far simpler. It is no longer necessary to carefully select which disks from which enclosures are assigned to a new RAID Group. Since all disks in a "stack" of enclosures attached to the same SAS wide cables are visible to the SAS controller chips, and because the enclosure "expanders" are intelligent, the SAS controller can choose which of the 4 SAS links to access the drives in every RAID Group in the most efficient manner. The SAS chips will dynamically load balance the disks across the four links, making changes as needed when adding new RAID Groups or after disk rebuilds for a failed disk. It is best to visualize these SAS back-ends as a matrix of disks to SAS links that are automatically adjusted for best performance by the controllers.

III. Cache Architecture

Basic Cache Operation Concepts

The default cache partitioning methods used on the AMS1000 Family and the AMS2000 Family are the same. Cache on each controller is automatically split into two regions at subsystem boot, creating a ***System Region*** and a ***User Data Region***.

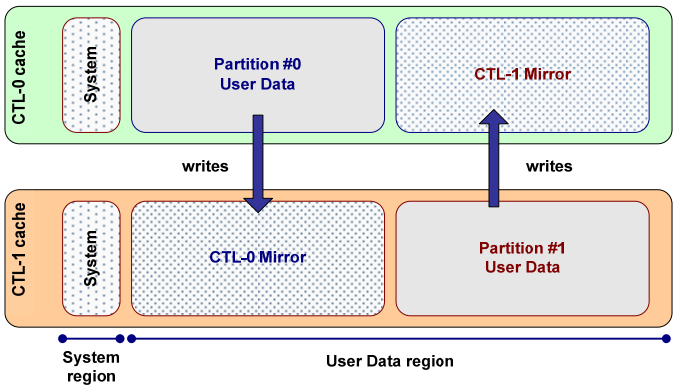


Figure 14. Cache configuration, no CoW or TCE software

System Region

The System Regions are automatically sized by each controller (CTRL) at boot time based on the actual hardware configuration. These System Regions will increase in size (after a reboot) if either *Copy-on-Write* (**CoW**) or T*rue Copy Extended* (**TCE**) licenses are installed (optional features, license key enabled), thus reducing the User Data Region by about 60%.

User Data Region

The User Data Region is equally split into a ***Data Area*** and a ***Mirror Area***. The *Data Area* is that space where all I/O requests are processed on a controller for those LUNs that it currently manages. This Data Area is at 50% of the total space available in the User Data Region. The name of this base Data Area on Controller-0 is ***Partition #0,*** and for Controller-1 it is ***Partition #1***. Without the use of the optional Cache Partition Management software (CPM), these base partitions will be the only two partitions available in an AMS2000 array (one per controller). [Note: CPM is provided at no cost with the "BOS-M" package.] Each controller uses its local Data Area for processing all I/O requests against those LUNs that they currently manage. The sizes of these default partitions are shown below in Table 2.

The *Mirror Area* on each controller is used by the DCTL chip for the mandatory Write block mirroring (duplexed writes). After receiving an inbound write request for a LUN it manages, a controller will also send the request and the data to the other controller via the DCTL's 2GB/s inter-controller communications bus for redundancy. The Mirror Area is fixed at 50% of the User Data Region. It is not a considered to be partition.

LUN Assignment

On the AMS2000 Family, when LUNs are created they are assigned to the base partition on the managing controller. Recall that when LUNs are created on the AMS2000 Family, they are automatically assigned in a round-robin fashion (cannot be disabled) to the global management list for the two controllers. The base partition is either Partition #0 or #1. All I/O requests for that LUN are processed in the base partition of the managing controller.

Write Pending Limits

There is a usage limit of 70% per controller and per Partition on the amount of cache space that may be used for write operations (thus 30% of the space in each Write-only Mirror will never be used). The 70% figure comes from a 30% limit for a "Middle Dirty" queue (newly received but unprocessed write requests) and 40% for a companion "Physical Dirty" queue (write requests that are in the process of being executed). Therefore, in a customer solution where sustained heavy write rates are expected, the 4GB DIMM option should be selected for configuring the maximum amount of cache (the 4GB DIMMs) in the array. This will create the largest Data Area possible in order to maintain high levels of performance.

Default Cache Details (no software)

Figure 14 above is an illustration of any AMS2000 array that does not have Copy on Write (CoW) or TrueCopy Extended (TCE) software installed. If, for example, this were a well-configured AMS2500 with **16GB** of cache, the overall System Region would be **2,904MB** (1,452MB per controller) and the overall User Data Region would be **13,480 MB** (6,740MB per controller).

Within these User Data Regions, there will be:

- **Partition #0** on CTRL-0 of **3,370MB** (the Data Area, 50% of the CTRL-0 User Data Region)
- CTRL-0 **Mirror** Area of 3,370MB (50% of the CTRL-0 User Data Region)
- **Partition #1** on CTRL-1 of **3,370MB** (the Data Area, 50% of the CTRL-1 User Data Region)
- CTRL-1 **Mirror** Area of 3,370MB (50% of the CTRL-1 User Data Region)

This means that the overall usable cache space for user data operations is **6,740MB** if there is at least a 50% Read workload component. However, if this subsystem routinely experienced a 100% write workload, only 70% of that overall Data Area, or 4,718MB, will be available for I/O requests due to the 70% write operations limit.

System and User Data Region Sizes without CoW or TCe

Table 2 shows the sizes available for the User Data Region when there is no CoW or TCE software installed. The various cache sizes are also shown in this table. Note that as the cache size increases, so does the footprint used for the System Region of cache.

System Type	Total cache (GB)	Total System space - no software (MB)	Total User Data Region (MB)	Total Data space (MB)	Total Mirror space (MB)
AMS2500	32	3,928	28,840	14,420	*14,420*
AMS2500	16	2,904	13,480	6,740	*6,740*
AMS2300	16	2,064	14,320	7,160	*7,160*
AMS2300	8	1,632	6,560	3,280	*3,280*
AMS2100	8	1,152	7,040	3,520	*3,520*
AMS2100	4	1,056	3,040	1,520	*1,520*

Table 2. Size of System and User Data regions, no software

Default Cache Details with Software

Looking at Figure 15 below, this is an illustration of a system that does have either Copy on Write (CoW) or TrueCopy Extended (TCE) software installed. If, for example, this were a well-configured AMS2500 with 16GB of cache, the overall System Regions would now be 11,104MB (5,552MB per controller) and the overall User Data Regions would be 5,280MB (2,640MB per controller).

Within these *User Data Regions*, there will be:

- **Partition #0** on CTRL-0 that is **1,320MB** (the Data Area, 50% of the CTRL-0 User Data Region)
- CTRL-0 **Mirror** Area of 1,320MB (50% of the CTRL-0 User Data Region)
- **Partition #1** on CTRL-1 of **1,320MB** (the Data Area, 50% of the CTRL-1 User Data Region)
- CTRL-1 **Mirror** Area of 1,320MB (50% of the CTRL-1 User Data Region)

This means that the overall cache space for user data operations is **2,640MB** if there is at least a 50% Read workload component. However, if this subsystem routinely saw a 100% write workload, only 70% of that overall Data Area, or 1,848MB, will be available for I/O requests due to the 70% write operations limit.

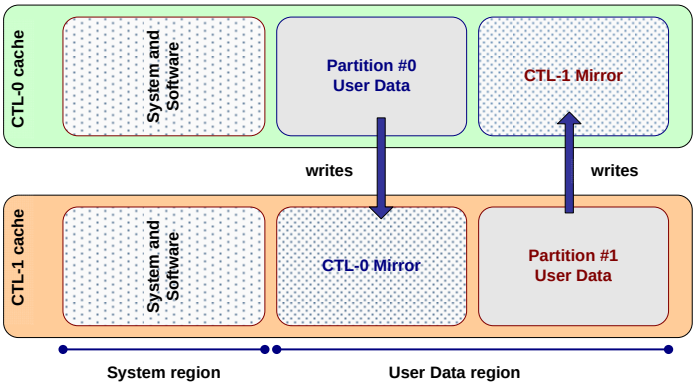


Figure 15. Cache configuration, with CoW or TCE software

System and User Data Region Sizes with CoW or TCe

Table 4 shows the sizes available for the User Data Region when either Copy on Write (CoW) or TrueCopy Extended TCE software has been installed. The various cache sizes are also shown in this table. Note that as the cache size increase, so does the footprint used for the System Region of cache.

System Type	Total cache (GB)	Total System space - with software (MB)	Total User Data Region (MB)	Total Data space (MB)	Total Mirror space (MB)
AMS2500	32	20,328	12,440	6,220	*6,220*
AMS2500	16	11,104	5,280	2,640	*2,640*
AMS2300	16	10,264	6,120	3,060	*3,060*
AMS2300	8	5,752	2,440	1,220	*1,220*
AMS2100	8	5,232	2,960	1,480	*1,480*
AMS2100	4	2,096	2,000	1,000	*1,000*

Table 3. Size of System and User Data regions, with software

Mirror Region Usage with a Controller Failure

In the event of a controller failure, what happens is that the reserved Mirror Area on the surviving controller is turned into a second Master Partition that manages all of the LUNs that were previously managed by the failed controller. Figure 16 shows how the cache in Controller-0 would look after a failure of Controller-1 (after some small preset period of time). All of the pending duplexed writes that were located in the local Mirror Region (duplexed writes from the other controller) would be destaged to disk immediately by Controller-0, and then it would become Partition-1.

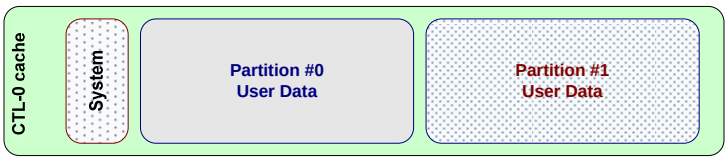


Figure 16. Cache configuration, with a controller failure

Cache Partition Manager Concepts

The configuration choices for cache when using the optional ***Cache Partition Manager*** (**CPM**) package are also almost the same for both the AMS1000 and AMS2000 Families. The basic discussion from the previous cache overview section also applies here.

Sub-partitions

When using CPM, there is a Master Partition established per controller (with a fixed 16KB cache segment size) and one or more user defined Sub-partitions. Each Sub-partition (but not the Master) may use alternate cache segment sizes. All of the LUNs managed by a controller default to using the Master Partition space. Only specifically selected LUNs (selected and assigned using the CPM utility) will use Sub-partitions, whereas the Master Partition is the default space for all LUNs which are managed by that controller.

Cache Segments

In addition to creating Sub-partitions, CPM can be used to set the cache segment sizes in each Sub-partition. A **cache segment** is the uniform unit of allocation from the cache system. One or more segments will be used as buffers for an I/O operation. The default segment size is 16KB, but alternate choices for individual Sub-partitions (if using CPM) can include 4KB, 8KB, 64KB, 256KB, and 512KB. These alternate sizes would only be used if the workloads against those LUNs assigned to that sub-partition would take advantage of a different segment size. An application that uses a large block size (512KB for example) and sequential I/O workloads would be such a candidate.

Master Partitions

As described in the previous section, the User Data Region contains all of the cache space not used by the System Region, and it is equally split into a ***Data Area*** and a ***Mirror Area*****.** The Data Area is that space where all I/O requests are processed on each controller. It is set as 50% of the total space available in the User Data Region. When using CPM, the name of this Data Area on CTRL-0 is ***Master Partition #0***, and for CTRL-1 it is ***Master Partition #1*****.** These default to 50% of the space available in the Data Area. The rest of the space is usable for assignment to user defined Sub-partitions.

Each Master Partition may be changed to be any size in between 200MB and the full size of the Data Area. Each Sub-partition may be any size from 100MB to the limit of remaining space in that Data Area. Note that it is a good idea to match these sizes across the controllers in order to manage smooth transitions during a controller LUN management change (Load Balancing enabled). When a LUN is assigned to a Sub-partition, and that Sub-partition doesn't exist on the other controller, the LUN will be reassigned to the Master Partition upon a controller management change. This could cause a very different performance behavior for those LUNs. If Sub-partitions are configured equally across the controllers, and use matching cache segment sizes, the LUNs that are moved will have their I/O requests processed in the matching Sub-partition on the other controller.

LUN Assignment

When LUNs are created they are assigned to the base partition on the managing controller. Recall that when LUNs are created on the AMS2000 Family, they are automatically assigned in a round-robin fashion to the global management list for the two controllers. When using CPM, the

base partition is either Master Partition #0 or #1. All I/O requests for that LUN are processed in the base partition of the managing controller.

Cache Segments and RAID Stripe Size Combinations

When using CPM, each Sub-partition may be configured to use a different cache segment size. On the AMS2000, when a LUN is created, the default ***RAID Group*** stripe size is set to 256KB. An alternate choice of a 64KB or 512KB stripe size per LUN is also available without the use of CPM. If CPM is installed, then alternate ***cache segment*** sizes may be used. These include 4KB, 8KB, 64KB, 256KB, or 512KB instead of the default 16KB. Previously, on the AMS1000 Family, CPM was required in order to change the cache segment size *or* RAID Group size.

The configuration software has interlocks that only permit certain combinations of RAID Group stripe size and cache segment size may be used together. Table 4 below shows these combinations.

	RAID Stripe Size		
Cache Segment Size	64KB (default for AMS1000 Family)	256KB (default for AMS2000 Family)	512KB
4KB	√	-	-
8KB	√	√	-
16KB (default)	√	√	√
64KB	√	√	√
256KB	-	√	√
512KB	-	-	√

Table 4. Combinations of RAID stripe size and cache segment size

System and User Data Region Sizes without CoW or TCe Software

Figure 17 below is an illustration of a system does not have either Copy on Write (CoW) or TrueCopy Extended (TCE) software licenses installed. If this were a well-configured AMS2500 with **16GB** of cache, the overall **System region** would be **2,904MB** (1452MB per controller) and the overall **User Data region** would be **13,480 MB** (6,740MB per controller).

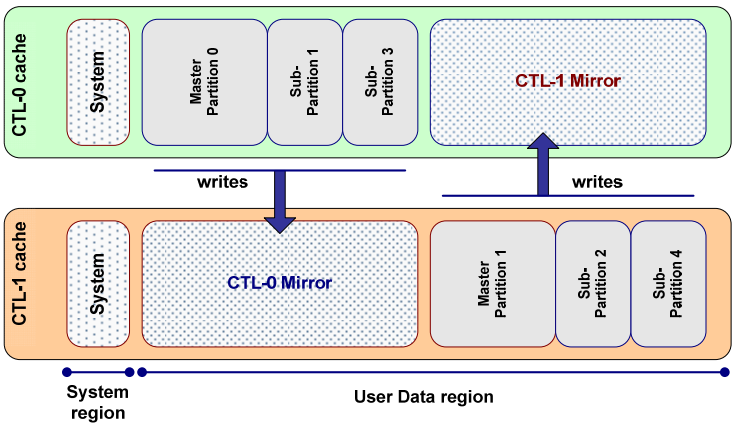


Figure 17. CPM Cache configuration example, not using CoW or TCe software

Within these User Data regions, the CPM default configuration will be:

- **Master Partition #0** on CTRL-0 of **1,685MB** (the Data Area, default of 50% of the CTRL-0 User Data Region) and **1,685 MB** available for sub-partitions.
- CTRL-0 **Mirror** area of 3,370MB (50% of the CTRL-0 User Data Region)
- **Master Partition #1** on CTRL-1 of **1,685MB** (the Data Area, default of 50% of the CTRL-1 User Data Region) and **1,685 MB** available for sub-partitions.
- CTRL-1 **Mirror** area of 3,370MB (50% of the CTRL-1 User Data Region)

Using the sub-partition layout example from Figure 17, if all sub-partitions were created to be the same size, then all four of them would be 842.5MB each. Note that the previously mentioned write pending limit is at 70% of the space per partition.

System and User Data Region Sizes with CoW or TCe Software

Figure 18 below is an illustration of a system that does have Copy on Write (CoW) or TrueCopy Extended (TCE) software installed. If this were a well-configured AMS2500 with **16GB** of cache, the overall **System region** would now be **11,104MB** (5,552MB per controller) and the overall **User Data region** would be reduced to only **5,280MB** (2,640MB per controller).

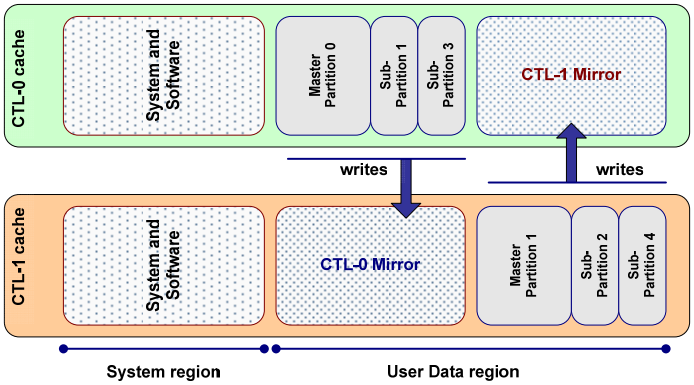


Figure 18. CPM Cache configuration example, using CoW or TCe software

Within these User Data regions, the CPM default configuration will be:

- **Master Partition #0** on CTRL-0 that is **1,320MB** (by default 50% of the CTRL-0 User Data Region) and 1,320MB available for sub-partitions.
- CTRL-0 **Mirror** region of 2,640MB (50% of the CTRL-0 User Data Region)
- **Master Partition #1** on CTRL-1 of 1,320MB (by default 50% of the CTRL-1 User Data Region) and 1,320MB available for sub-partitions.
- CTRL-1 **Mirror** region of 2,640MB (50% of the CTRL-1 User Data Region)

This leaves 1,320MB of space on each controller from which to create one or more *Sub-partitions*. In this example, the Master Partition sizes may be adjusted to any size between 200MB and 2,440MB (200MB less than the full User Data space of 2,460MB per controller). One or more Sub-partitions may be independently created on each CTRL from the unused space in each User Data area. The number of Sub-partitions defined per system varies by model and

cache size. Each Sub-partition may be any size from 100MB up to the limit of the free space in that controller's User Data area.

IV. General Storage Concepts

Understand Your Customer's Environment

Before recommending a storage design for customers, it is important to know how the product will address the customer's specific business needs. The factors that must be taken into consideration include: ***Capacity, Performance, Reliability, Features and Cost*** with respect to the storage infrastructure component. The more data you have, the easier it is to architect a solution as opposed to just selling another storage unit. The types of storage configured, including the disk types, the number of RAID Groups and their RAID levels, as well as the number and type of host paths is important to the solution.

Disk Types

Table 5 below lists the disks types available in the AMS2000 storage arrays. Note that the table lists the true sizes typically seen by a host after formatting. The additional formatting applied by host-based Logical Volume Managers and file systems will consume additional space.

HDD Type	Advertised Size (GB)	Actual Capoacity (base 10)	Actual Capoacity (base 2)	Typical Physical max IOPS
1000 GB SATA 7.2k RPM	1000	983.7	916.1	80
400 GB SAS 10k RPM	400	392.73	365.8	130
300 GB SAS 15k RPM	300	287.63	267.9	180
146 GB SAS 15k RPM	146	143.31	133.5	180

Table 5. List of disks types and characteristics

Table 5 illustrates the advertised size (base-10), the actual (base-10) raw size, and the typical usable (formatted base-2) size of each type of disk. Also shown is the "rule of thumb" average maximum random 4K IOPS rate for each type of disk when using most of the surface. Every disk has a maximum random small block IOPS rate determined by seek times (head movement across the disk) and rotational delays (disk RPM). The number and size of the LUNs created per RAID Group determine how much of a disks' surface is in active use. When using the full disk surface, any expectations for higher sustained levels of IOPS per disk must be met by sustained cache hit rates on the storage array (avoiding physical reads from disk). Note that 10-20% is a typical cache hit rate on Open Systems servers.

As more of the disk surface is allocated to LUNs for a RAID Group, the farther the disk heads must seek. This creates higher average seek times. For example (using a 15k RPM disk), when only using the outer 25% of the surface the average read seek time can be around 1.8ms (providing about 267 IOPS), whereas the 100% surface average seek time will be about 3.8ms (about 182 IOPS). For a SATA disk at 7200 RPM, these values are about 4.3ms (119 IOPS) for using 25% of the disk surface and 8.5ms (79 IOPS) when using the full disk surface. All write rates are about 5% lower than the read rates due to the higher seek precision required. See Appendix F for more details on theoretical estimates of disk IOPS rates.

RAID Levels

The AMS2000 Family of storage systems currently supports RAID levels 1, 5, 6, and 10. RAID-5 is the most space efficient of these four RAID levels.

RAID-1 is a simple mirroring of two disks. Though this won't provide much in the way of IOPS or throughput because of the small number of disks, it could be useful for some logs or even system disks such as needed by VMware installations.

RAID-5 is a group of disks (typically referred to as either a RAID Group or Array Group) with the space of one disk used for the rotating parity chunk per RAID stripe (row of chunks across the set of disks). If using a 7D+1P configuration (7 data disks, 1 parity disk), then you get 87.5% capacity utilization for user data blocks out of that RAID Group.

RAID-6 is RAID-5 with a second parity disk for a second unique parity block. The second parity block includes all of the data chunks plus the first parity chunk for that row. This would be indicated as a 6D+2P construction (75% capacity utilization) if using 8 disks.

RAID-10 is a mirroring and striping mechanism. First, individual pairs of disks are placed into a mirror state. Then, all of these pairs are used in a simple RAID-0 stripe. If using 8 disks in the RAID Group, this would be represented as RAID-10 (4D+4D) and have 50% capacity utilization. RAID-10 is not the same as RAID-0+1, although sloppy usage by many would lead one to think this is the case. RAID-0+1 is a RAID-0 stripe of N-disks that is mirrored to another RAID-0 stripe of N-disks. This would also be shown as 4D+4D for an 8 disk construction. However, if one disk fails, that RAID-0 stripe also fails, and the mirror then fails, leaving a user with a single unprotected RAID-0 group. In the case of real RAID-10, one disk of each mirror pair would have to fail before getting into this same unprotected state.

The factors in determining which RAID level to use are cost, reliability, and performance. Table 6 shows the major benefits and disadvantage of each RAID type. Each type provides its own unique set of benefits so a clear understanding of your customer's requirements is crucial in this decision.

	RAID-1 / RAID-10	RAID-5	RAID-6
Description	Mirroring / Data Striping and Mirroring	Data Striping with distributed parity	Data Striping with two distributed parities
Minimum # Disks	2/4	3	4
Maximum # Disks	2/16	16	30
Benefit	Highest performance with data redundancy	The best balance of cost, reliability, and performance.	Balance of cost, with extreme emphasis on reliability
Disadvantages	Higher cost per number of physical disks	Performance penalty for high percentage of Random Writes	Performance penalty for all writes

Table 6. RAID levels comparison.

Another characteristic of RAID is the idea of "write penalty". Each type of RAID has a different back-end physical disk I/O cost, determined by the mechanism of that RAID level. The Table below illustrates the trade-offs between the various RAID levels for write operations. There are additional physical disk reads and writes for every application write due to the use of mirrors or XOR parity. SATA disks are often deployed with RAID-6 to protect against a second drive failure within that RAID Group during the lengthy disk rebuild of a failed drive.

SAS	Array IOPS per Host Read	Array IOPS per Host Write
RAID10	1	2
RAID5	1	4
RAID6	1	6
SATA-II		
RAID10	1	4
RAID5	1	6
RAID6	1	9

Table 7. Tables of RAID level write penalties.

With the AMS1000 and AMS2000 Families of array, when using SATA disks a ***read verify*** is used after each write of either data or parity. This mechanism verifies that the SATA disk wrote the data where it was told to write the data. This command is not used for SAS disks. This means that, in the case of SATA disks, there are additional physical I/O operations per host write in order to perform this read verify operation. With RAID-6, there are three such blocks: data, parity1, and parity2. Since this verification is performed within the SATA disk canister logic, it is much more efficient that it was on the AMS1000 Family, where the DCTL processor managed these verifications.

	IOPS Used
RAID-10	1 Data Write, 1 mirrored Data Write
RAID-5	2 reads (1 data, 1 parity), 2 writes (1 data, 1 parity)
RAID-6	3 reads (1 data, 2 parity), 3 writes (1 data, 2 parity)
	SATA writes: also read and compare the data and parity writes

Table 8. Break down of RAID level write costs (FC disks)

RAID Groups and Parity Groups

On Hitachi midrange products the terms **RAID Group** and **Parity Group** are used when configuring storage. Parity Group is the actual RAID element – such as the four-disk RAID-10 (2D+2D). Usually the term Parity Group and RAID Group will refer to the same thing. However, an AMS array also allows for the creation of a RAID Group that can span several Parity Groups. As this is a *concatenation* and not a single large RAID stripe, it is not a recommended practice due to the potential performance issues.

RAID Chunks and Stripes

A RAID Group is a logical mechanism that has two basic elements: a virtual block size from each disk (a **chunk**) and a row of chunks across the group (the RAID **stripe**). The chunk size is typically set to 64KB on midrange arrays, but is adjustable (64KB, 256KB, 512KB) on Hitachi AMS arrays. It is adjustable on the AMS1000 Family, but only if the optional Cache Partition Management (CPM) package is installed. The use of CPM allowed for several choices of chunk size as well as

cache slot size. On the AMS2000 Family, the RAID chunk size defaults to 256KB, and is adjustable **to other values (per LUN) via the Storage Navigator Modular management tool. It does not** require the installation of the CPM package.

The stripe size is the sum of the chunk sizes across a RAID Group. This only counts the "data" chunks and not any mirror or parity space. Therefore, on a RAID-6 group created as 8D+2P (ten disks), the stripe size would be 512KB (64KB chunk) or 2KB (256KB chunk).

Note that some usage replaces *chunk* with "stripe size", "stripe depth", or "interleave factor", and *stripe size* with "stripe width", "row width" or "row size".

Note that on all current RAID systems, the chunk is the primary unit of protection management: either the parity or mirror mechanism. I/O is not performed on a chunk basis as is commonly thought. On Open Systems, the entire space presented by a LUN is a contiguous span of 512 byte blocks, known as the Logical Block Address range (LBA). The host application makes I/O requests using some native request size (such as a file system block size), and this is passed down to the storage as a unique I/O request. The request has the starting address (of a 512 byte block) and a length (such as the file system 8KB block size). The storage array will locate that address within that LUN to a particular disk address, and read or write only that amount of data – not that entire chunk. Also note that this request could require two disks to satisfy if 2KB of the block lies on one chunk and 6KB on the next one in the stripe.

Because of the variations of file system formatting and such, there is no way to determine where a particular block may lie on the raw space presented by a volume. A file system will create a variety of metadata in a quantity and distribution pattern that is related to the size of that volume. Most file systems also typically scatter writes around within the LBA range – an outdated hold-over from long ago when file systems wanted to avoid a common problem of the appearance of bad sectors or tracks on disks. What this means is that attempts to align application block sizes with RAID chunk sizes is a pointless exercise.

The one alignment issue that should be noted is in the case of host-based Logical Volume Managers. These also have a native "stripe size" that is selectable when creating a logical volume from several physical storage LUNs. In this case, the LVM stripe size should be a multiple of the RAID chunk size due to various interactions between the LVM and the LUNs. One such example is the case of large block sequential I/O. If the LVM stripe size is equal to the RAID chunk size, then a series of requests will be issued to different LUNs for that same I/O, making the request appear to be several random I/O operations to the storage array. This can defeat the array's sequential detect mechanisms, and turn off sequential prefetch, slowing down these types of operations.

LUNS (host volumes)

On a midrange system, when space is carved out of a RAID Group and made into a volume, it is then known as a Logical Unit (LU). Once the LU is mapped to a host port for use by a server, it is known as a LUN (Logical Unit Number) and is assigned a certain World Wide Name if using fibre channel interfaces on the array. On an iSCSI configuration, the LUN gets a name that is associated with an NFS mount point. Note that general usage has turned the term of "LU" into "LUN".

Number of LUNs per RAID Group

When configuring a midrange array, one or more LUNs can be created per RAID/Array Group, but the goal should be to clearly understand what percentage of that group's overall capacity will contain active data. In the case where multiple hosts attempt to simultaneously use LUNs that share the same physical disks in an attempt to fully utilize capacity, seek and rotational latency may be a performance limiting factor. In attempting to maximize utilization, RAID groups should contain both active and less frequently used LUNs. This is true of all physical disks regardless of size, RAID level, and physical characteristics.

It is also true that, if many small LUNs are carved out of a single RAID group, their simultaneous use will create maximum seek times on each disk, reducing the maximum sustainable small block random IOPS rate to the disk's minimum.

LUN Management and Controller I/O Management

On nearly every midrange storage array from any vendor, the individual LUNs are tightly bound to an "owning" controller. This is because there is no global sharing between the controllers of either the data or its metadata. Each controller is independently responsible for managing these two objects. On enterprise arrays, there is no concept of either a "controller" or "LUN ownership". All data and metadata on an enterprise system is globally shared by all front-end processors.

AMS2000 Family: LUN management

As mentioned earlier, the AMS2000 Family introduces a totally new concept to the midrange array arena. The rigid concept of LUN ownership by controller has been replaced with a more enterprise-like method of LUN Management. Rather than simple ownership, now there is a global table of all LUNS that determines which controller will execute any I/O request for a specific LUN. This is independent of which host port on which controller is involved in the I/O request. The Active/Active Symmetric front-end design enables this new capability and the corresponding freedom from micro-managing the appearance of LUNs on certain paths for certain hosts. All LUNs are assigned on a round-robin basis to a controller's I/O management list as they are created. The table of I/O management is changed by the operation of the previously described Hardware Load Balancing feature that remaps, over time, certain LUNS to the alternate controller.

On many other midrange arrays, LUN ownership is by the controller to which that LUN was originally assigned when the LUN was created. This ownership creates a reference for that controller to manage both the cache hit/miss determination for Random I/O requests and for cache mirroring of write blocks (duplexed writes on Hitachi midrange arrays). On the AMS1000 Family, a controller may access LUNs owned by the other controller under either of two conditions: (1) Data Share mode and (2) Ownership Change. In the case of controller failure, the functioning controller will immediately take control of all LUNs previously managed by the other controller. See Appendix E for a more detailed description.

Port I/O Request Limits, LUN Queue Depths, and Transfer sizes

There are three aspects of I/O request handling per storage path that need to be understood. At the port level, there is the mechanism of an **I/O request limit** and a **maximum transfer size**. At the LUN level, there is the mechanism of the **maximum queue depth**.

Port I/O Request Limits

Each server and storage array has a particular ***port I/O request limit***, this being the number of total outstanding I/O requests that may be issued over a path at any point in time. This limit will vary according to the server's Fibre Channel HBA and its driver, as well as the limit supported by the target storage port. Switches serve to aggregate many HBAs to the same storage system port. The operating limit will be the lower of these two points. In the case of a SAN switch being used to attach multiple server ports to a single storage port (fan-in), the limit will most likely be the limit supported by the storage port.

On Hitachi midrange arrays, the I/O request limit per port is 512. This means that, at any one time, there can be up to 512 active host(s) I/O commands queued up for the various LUNs visible on that port. If the number of active LUNs is high enough, then this port limit will throttle the maximum queue depth per LUN. For example, if there are 24 LUNs mapped to a port, and all of them are equally busy with small block random workloads, then the average queue depth per LUN (SAS disks) will tend to be about 21.

Most environments don't require that all LUNs be active at the same time. As I/O requests are application driven, this must be taken into consideration. Understanding the customer's requirements for performance will dictate how many LUNs should be assigned to a single port. Environments that have low I/O demands overall will allow a large number of LUNs to be assigned per port without routinely exceeding the queue limit for that port. On the other hand, environments with highly active devices and multiple I/O request should be designed with fewer LUNs per port, where additional ports will be required in order to satisfy the I/O demand.

Port I/O Request Maximum Transfer Size

There is also a host tunable parameter at the driver level that controls the maximum transfer size of a single I/O operation. This is usually a per-port setting, but it also might be a per-host setting. The defaults for this can range from fairly small (like 32KB) to something midsized (128KB). The maximum is usually 1024KB. It is probably a best practice to set this to the largest setting that is supported on the host. The maximum transfer size that an AMS will accept is 1024KB. This value controls how application I/O requests get processed. If one used a large application block size (say 256KB) and the port/LUN default was just 32KB, then each request would be broken down (fragmented) into 8 * 32KB requests. This creates additional overhead in managing the application request. The use of a large maximum transfer size such as 1024KB will often be readily apparent on the performance of the system.

LUN Queue Depth

At the LUN level, there is the mechanism of the ***maximum queue depth***. This is the limit of how many outstanding I/O requests may be scheduled by a host for each LUN. On a host HBA, the device queue depth limit is usually configurable as a driver tunable, usually on a per-port basis

rather than a global setting. On a storage system, this limit is imposed by the controller and is not configurable. The usable queue depth per LUN when there are multiple simultaneously active LUNs from the same RAID group limit will be determined by physical I/O capacity of that group of disks.

On the AMS2000 Family, the recommended queue depth limits per LUN are **32 for SAS** RAID Groups and **16 for SATA-II** RAID Groups. As a comparison, on the AMS1000 Family, the LUN queue depth limit is different for LUNs using FC disks or SATA disks. On the AMS1000 Family, LUNs from RAID Groups that use FC disks have a maximum queue depth of 32, whereas a LUN from a RAID Group using SATA disks has a maximum queue depth of 4.

It is important to note that the effective queue depth limit per LUN is not directly related to the host I/Os as such, but more to the actual physical disk operations required for that RAID level (see Table 7 above), and for the alignment of the data blocks requested. For example, on a read, it is possible for an application block (say 8KB) to occupy space on two adjacent chunks in the RAID group, so there will be two disk reads for that one host read. For writes, not only is this true, but there is also the RAID level write penalty to add. For a write where the block is entirely on one RAID chunk, then there will be 2 (RAID-10), 4 (RAID-5) or 6 (RAID-6) physical I/Os if using FC or SAS disks. In the case of SATA disks, this becomes 4 (RAID-10), 6 (RAID-5) or 9 (RAID-6). All of these physical operations will double if the data alignment crosses RAID chunk boundaries. Note that the default RAID chunk size is 256KB on the AMS2000 Family and 64KB on the AMS1000 Family.

Mixing Data on the Physical Disks

Physical placement of data by RAID Groups (not just the LUNs from the same RAID Group) is extremely important when the data access patterns differ. Mixing highly "write intensive" data with high "read intensive" data will cause both to suffer performance degradation. This performance degradation will be much greater when using RAID-5 or RAID-6 due to the increase in back-end disk operations required for writes. When using FC and SAS disks, there are two physical I/Os required for each random write for RAID-10, four for RAID-5, and six for RAID-6. When using SATA disks and RAID6, there are three additional physical I/Os required due to the read verification after writing (data and parity/parity).

Workload Characteristics

Most applications, from a disk subsystem point of view, can be categorized into one of seven types of application profiles (though there are the extreme cases that are not covered here):

- Online Transactional Processing – Random Read/Write to data files, with Sequential Read/Write to Log functions
- Decision Support Systems – Sequential or Random Read, depending on data access method, post ETL
- General Purpose File systems – Random Read/Write with potentially some Sequential I/O
- Messaging systems – Random Read/Write
- Web Server systems – Mixed Random and Sequential with various block sizes.
- Content Rich Media Streaming and/or Video Streaming – Sequential Read/Write
- High-performance Computing – Sequential Read/Write

The applications that do not directly fit into one of these seven categories will most likely be a combination of two or more of these categories. Some environments underneath the applications, such as VMware and other server virtualization products, will also cause different patterns of behavior.

Selecting the Proper Disk Drive Form Factor

In all cases, distributing a workload across a higher number of small-capacity, high-RPM drives will provide better performance in terms of full random access. Even better results can be achieved by distributing the workload over a higher number of small LUNs where the LUNs are the only active LUNs in the Raid Group. When cached data locality is low, multiple, small-capacity, high-RPM drives should be used.

One must also take into consideration the case where a system that is only partially populated with 15k RPM disks will be able to provide a much higher aggregate level of host IOPS if the same budget is applied to lower cost 10k RPM disks. If there is a 50% increase in the cost of the 15k disks, then one could install 50% more disks of the 10k variety. The individual I/O will see some increase in response time when using 10k disks, but the total amount of IOPS available will be much higher.

Mixing I/O Profiles on the Physical Disks

Mixing large-block sequential I/O with small-block random I/O on the same physical disks can result in poor performance. This applies to both read and write I/O patterns. This problem can occur at the LUN level or RAID Group level. In the first case, with a single large LUN, files with different I/O profiles will result in poor performance due to lack of sequential detect for the large block sequential I/O requests. In the second case, where multiple LUNs share a RAID Group, files having different I/O profiles will result in sequential I/O dominating the disk access time, due to pre-fetching, thereby creating high response times and low IOPS for the small-block random I/O. The resolution to these two issues is to use different LUNs and different parity groups for different data types and I/O profiles.

Front-end Port Performance and Usage Considerations

The flexibility of the front-end ports is such that several types of connections are possible. A couple of port usage points are considered:

- Port fan-in and fan-out
- I/O profile mixes on a port

Host Fan-in and Fan-out

With a SAN, the ability to share a port or ports among several host nodes is possible, as is configuring a single host node to multiple storage ports. Fan-in is one of the great promises of fiber channel based SAN technologies—the sharing of costly storage resources. Several host nodes are fanned *into* a single storage port, or a host node has fanned out to several storage ports.

Fan-in

Host Fan-in refers to the consolidation of many host nodes into one or just a few storage ports (many-to-one). Fan-in has the potential for performance issues by creating a bottleneck at the front-end storage port. Having multiple hosts connected to the same

storage port does work for environments that have minimal performance requirements. In designing this type of solution, it is important to understand the performance requirements of each host. If each host has either a high IOPS or throughput requirement, it is highly probable that a single 2-gigabit or 4-gigabit FC-AL port will not satisfy their aggregate performance requirements.

Fan-out

Fan out allows a host node to take advantage of several storage ports (and possibly additional port processors) from a single host port (one-to-many). Fan-out has a potential performance benefit for small block random I/O workloads. This allows multiple storage ports (and their queues) to service a smaller number of host ports. Fan-out typically does not benefit environments with high throughput (MB/sec) requirements due the transfer limits of the host bus adapters (HBAs).

Mixing I/O Profiles on a Port

Mixing large-block sequential I/O with small-block random I/O can result in poor performance. This applies to both read and write I/O patterns. This problem can occur at the LUN level or RAID Group level. In the first case, with a single large LUN, files with different I/O profiles will result in poor performance due to lack of sequential detect for the large block sequential I/O requests. In the second case, where multiple LUNs share a RAID Group, files having different I/O profiles will result in sequential I/O dominating the disk access time, due to pre-fetching, thereby creating high response times and low IOPS for the small-block random I/O. The resolution to these two issues is to use different LUNs and different parity groups for different data types and I/O profiles.

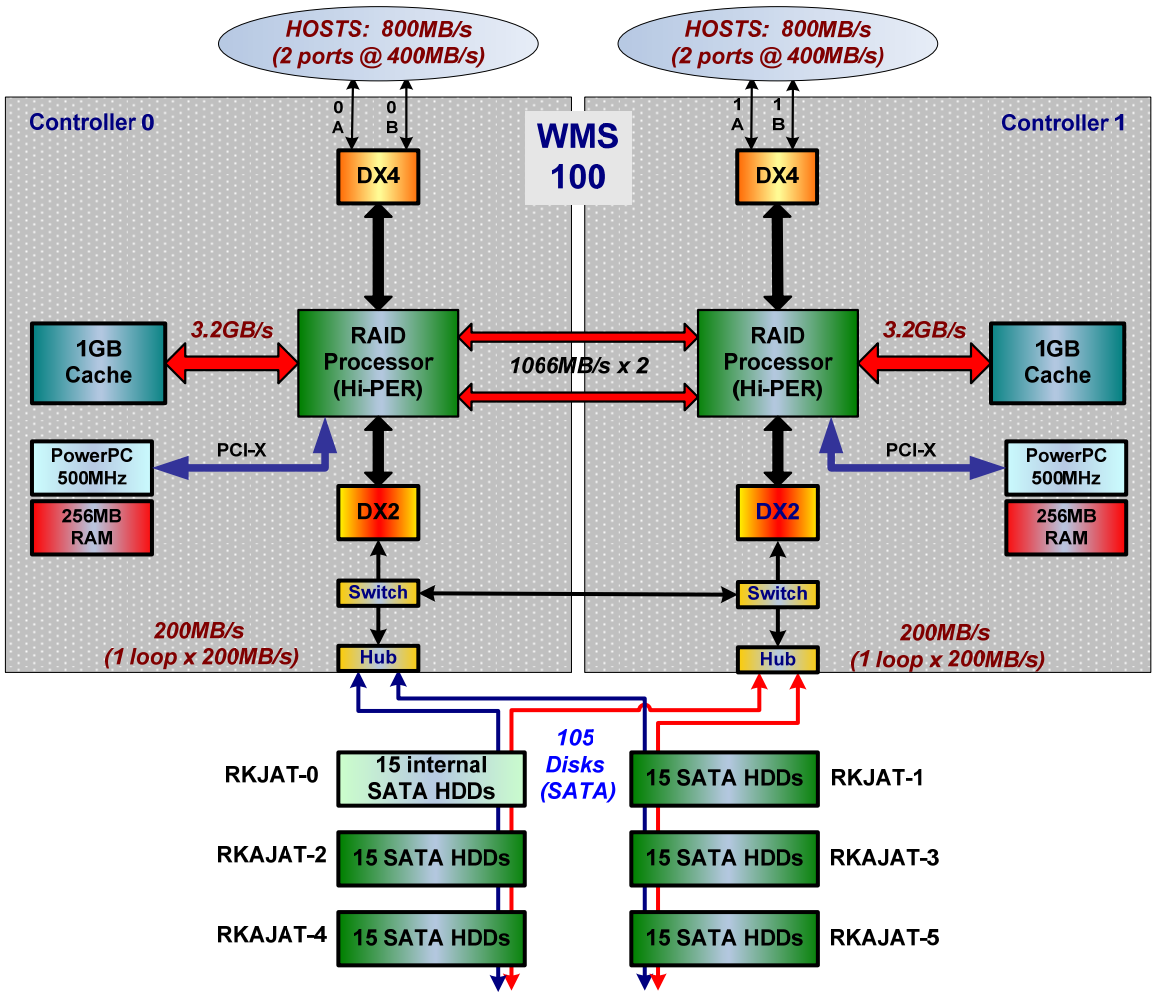
VI. Summary

The AMS2000 models offer great flexibility and they should perform extremely well in any number of environments. It will be important that Hitachi Data Systems sales personnel, technical support staff, value-added resellers, and others who are responsible for the delivery of solutions fully understand the concepts presented in this paper. Due to the considerable changes in the architecture and operation of these arrays, all of these people need to understand why they cannot simply carry over AMS1000 Family knowledge or implementation practices. As is true of all storage solutions, they must also invest the time required to design the best possible solution to meet each customer's unique requirements, whether it be capacity, reliability, performance or cost.

The following are key elements to successful solutions delivery.

- A clear understanding of your customer's environment or planned environment for net new installations. Completing the systems assurance document (SAD) prior to installation will help to ensure this.
- Set the proper performance expectations. Don't expect high performance if consolidating multiple applications on the same storage unit. Knowing your customer's workload and clearly understanding all aspects will avoid costly mistakes in overselling the capabilities of these products.
- Select the proper RAID Level to meet the I/O requirements. A focus on capacity alone without taking into consideration the Read/Write ratio can be costly to remedy after the sale.
- Clearly understand how much active data will reside on each array group. Installing large capacity drives to meet a price point, for applications that have a very low cache hit rate, may lead to performance issues and dissatisfied customers.
- Clearly understand how many hosts will share front-end ports, back-end disks and the specific requirements for each. It is possible to share ports and disks between low-access requirements and high-performance requirements, but without understanding the requirements, these products may be poorly configured and performance issues may arise that could have easily been avoided.
- When in doubt, engage Hitachi Data Systems Global Solution Services (GSS) organization or your local reseller to assist in gathering data regarding existing storage units.

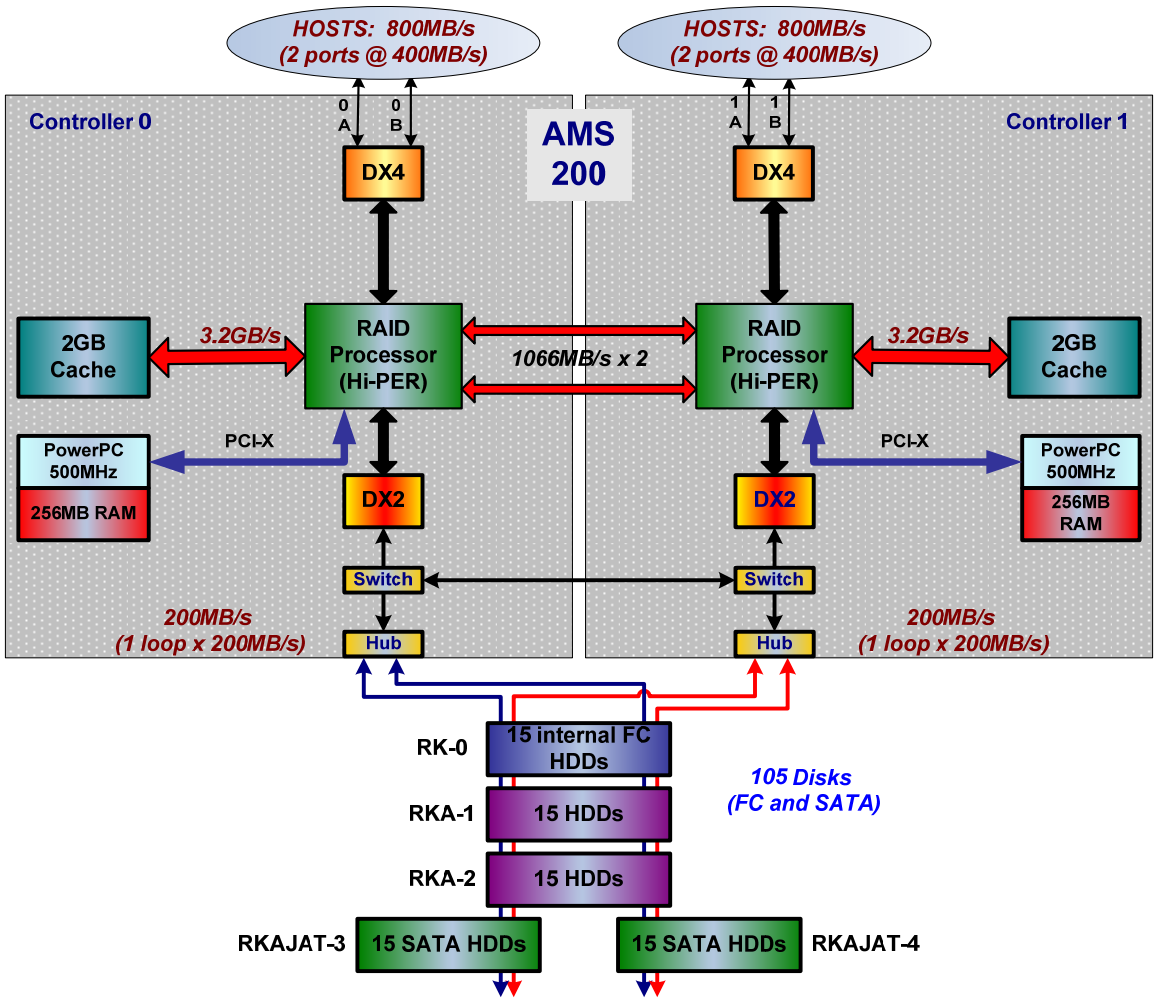
Appendix A. WMS100 Architecture



The WMS100 provides 4 4Gbit/s FC host paths, up to 2GB of cache, and up to 105 SATA-I disks. There are 5-15 disks in the internal drive bays. The controller processor is a PowerPC, and the RAID processor is the DCTL-S.

Notice how there is one back-end 2Gbit/s path from each controller on the WMS100. There are two connections (two pairs of IN-OUT connections per controller), but these are actually the same path as they are connected internally by a hub. There is a switch to provide for fail-over between the controllers. Only SATA enclosures may be installed on this system.

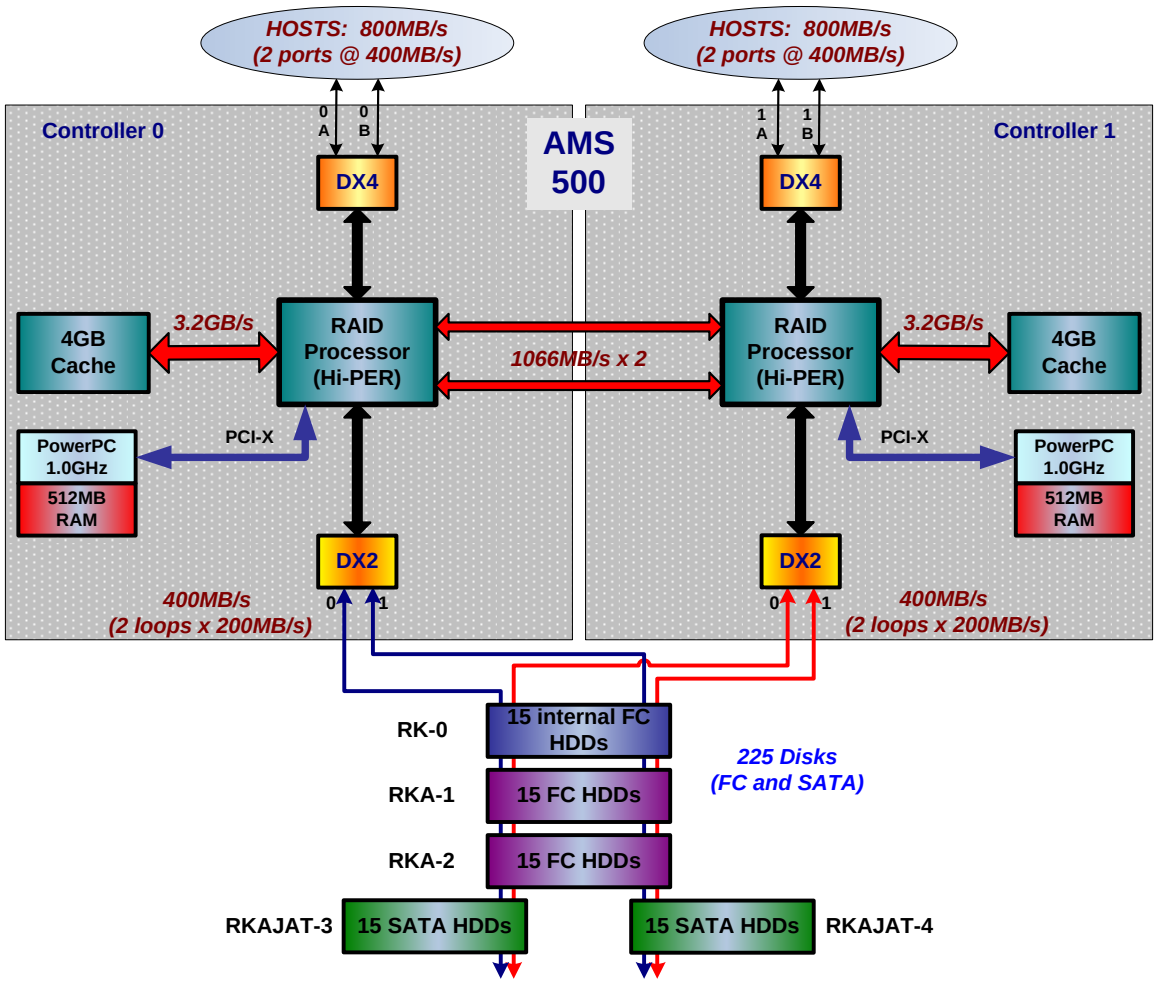
Appendix B. AMS200 Architecture



The AMS200 provides 4 4Gbit/s host FC paths, up to 4GB of cache, and up to 105 FC and/or SATA disks. There are 5-15 disks in the internal drive bays. There is a different enclosure type for the two kinds of disks. The controller processor is a PowerPC, and the RAID processor is the DCTL-S.

Notice how there is one back-end 2Gbit/s path from each controller on the AMS200. There are two connections (two pairs of IN-OUT connections per controller), but these are actually the same path as they are connected internally by a hub (sometimes referred to as the PCB). There is a switch to provide for fail-over between the controllers. Both FC and SATA enclosures may be installed on this system.

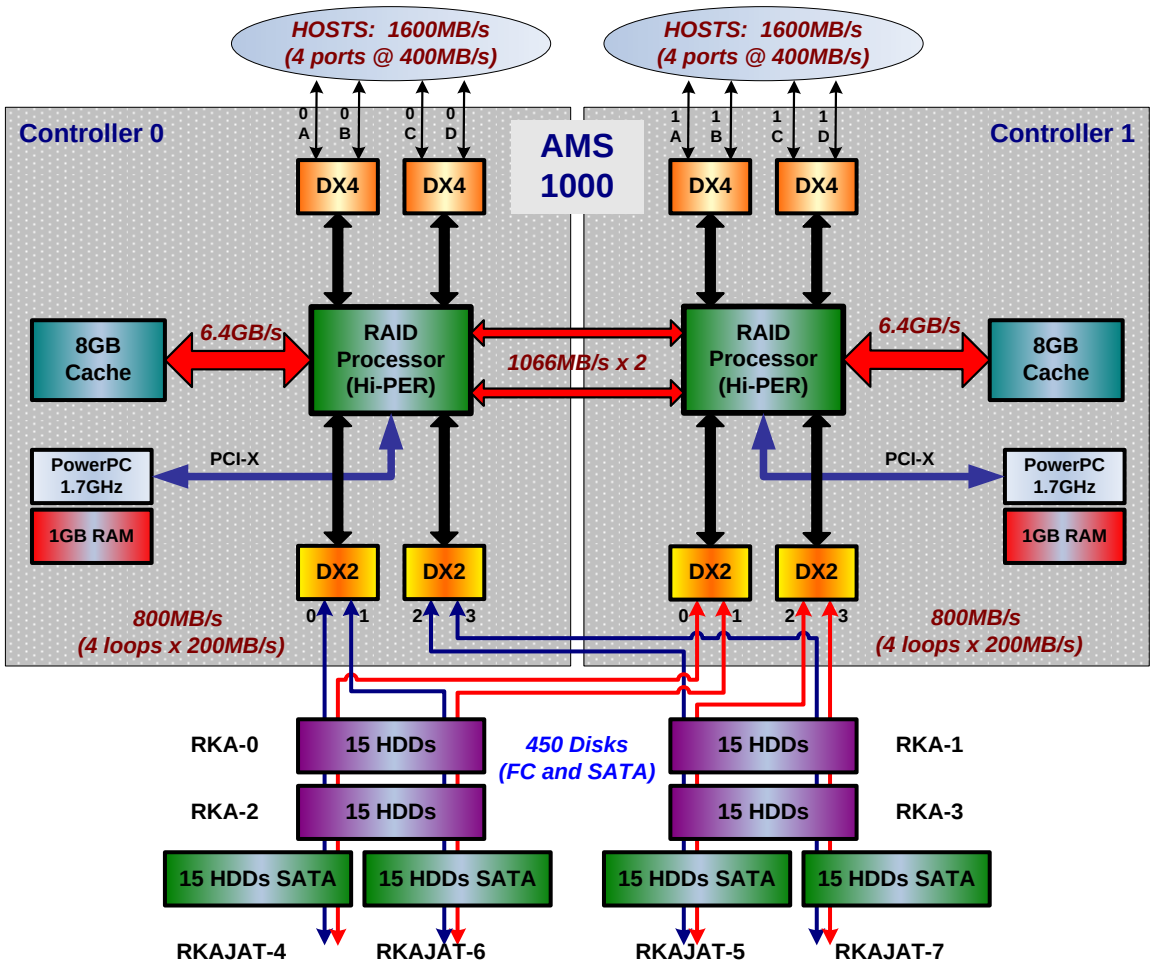
Appendix C. AMS500 Architecture



The AMS500 provides 4 4Gbit/s host FC paths, up to 8GB of cache, and up to 225 FC and/or SATA disks. There are 5-15 disks in the internal drive bays. There is a different enclosure type for the two kinds of disks. The controller processor is a PowerPC, and the RAID processor is the DCTL-H.

The AMS500 has two independent back-end 2Gbit/s paths from each controller. There are two connections (two pairs of IN-OUT connections per controller). As there are two active back-end paths per controller, all disks can be seen by just one controller in the event of a failure of the alternate controller. Both FC and SATA enclosures may be installed on this system.

Appendix D. AMS1000 Architecture



The AMS1000 provides 8 4Gbit/s FC paths, up to 16GB of cache, and up to 450 FC and/or SATA disks. There are no internal drive bays. There is a different enclosure type for the two kinds of disks. The controller processor is a PowerPC, and the RAID processor is the DCTL-H.

The AMS1000 has four independent back-end 2Gbit/s paths from each controller. There are two connections (two pairs of IN-OUT connections per controller). As there are four active back-end paths per controller, all disks can be seen by just one controller in the event of a failure of the alternate controller. Both FC and SATA enclosures may be installed on this system.

Appendix E. AMS1000 Family and Data Share mode

Midrange Arrays: LUNs and Controller Ownership

On both the Thunder 9500 series and AMS1000 Family,, individual LUNs are managed by an "owning" controller. This is also true of every other midrange product on the market. Initially, this is the controller that the LUN was originally assigned to when it was created. This ownership creates a reference for that controller to manage both the cache hit/miss determination for Random I/O requests and for duplexed write operations. A controller may access LUNs owned by the other controller under either of two conditions: (1) Data Share mode and (2) Ownership Change. In the case of controller failure, one controller will immediately take control of all LUNs previously managed by the other controller.

Data Share mode or Ownership Change

Available on later microcode releases of Thunder products (95xx series) and on all AMS1000 Family products, this feature addresses the issue of a LUN's ownership when accessed by both controllers. Assume Controller-0 (C0) is the original owning controller of a RAID Group and LUN-1 and LUN-2. Controller-1 (C1) has a host port configured that has also has access to LUN-2. Assume that LUN-2 is used by two hosts as a "raw" volume (no host-based file system) for an Oracle RAC environment.

When C1 needs to process an I/O to LUN-2, there are two possibilities that may occur. If C0 has not used LUN-2 for more than 60 seconds, then C0 will release ownership of LUN-2 over to C1. At that point C1 will begin directly process I/Os to LUN-2.

If, however, C0 has been processing I/O to LUN-2 within the past 60 seconds, LUN ownership will not change. What will happen is that C1 will pass the I/O request to C0 over the inter-DCTL processor bus for it to process. For write operations, C1 will copy the blocks to update from its own local cache into the C1 mirror region on C0, and into the local cache on C0 (as if it had received it from an external host). C0 will then process the "shared" write request, and copy the same blocks for this write into the C0 mirror region on C1. Thus, there are four copies of these blocks in all four areas of cache (local and mirrors).

Appendix F. Disks - Physical IOPS Details

These are examples of maximum expected physical disk Read and Write IOPS rates for three types of disks. The "Cylinders" columns show the percentage of surface space in use as disk partitions. As more disk surface is brought into use, the heads must seek farther inwards, which increases the average seek times. Note that the Write table shows reduced maximum IOPS due to the higher track centering precision required for writes. Please note in the tables below that the three drive types use a representative capacity. For instance, almost all 15K RPM HDDs would be very similar to the 300 GB Seagate 15K used in the first entry.

READS - 100% busy, high RT's	Cylinders			
300 GB Seagate FC 15k rpm	25%	50%	75%	100%
Disk rpm:	15,000	15,000	15,000	15,000
Average Read Seek Time:	1.8	2.5	2.8	3.5
Average rotational delay:	2.0	2.0	2.0	2.0
Average access time:	3.8	4.5	4.8	5.5
Random 8KB IOPS rate:	267	225	208	182
146 GB Hitachi FC 10k rpm	25%	50%	75%	100%
Disk rpm:	10,000	10,000	10,000	10,000
Average Read Seek Time:	2.4	3.3	3.8	4.7
Average rotational delay:	3.0	3.0	3.0	3.0
Average access time:	5.4	6.3	6.8	7.7
Random 8KB IOPS rate:	187	159	148	130
750 GB Seagate SATA 7.2k rpm	25%	50%	75%	100%
Disk rpm:	7,200	7,200	7,200	7,200
Average Read Seek Time:	4.3	6.0	6.8	8.5
Average rotational delay:	4.2	4.2	4.2	4.2
Average access time:	8.4	10.1	11.0	12.7
Random 8KB IOPS rate:	119	99	91	79

Table 9. Calculated maximum disk Read IOPS rates by type and active surface usage

WRITES - 100% busy, high RT's	Cylinders			
300 GB Seagate FC 15k rpm	25%	50%	75%	100%
Disk rpm:	15,000	15,000	15,000	15,000
Average Write Seek Time:	2.0	2.8	3.2	4.0
Average rotational delay:	2.0	2.0	2.0	2.0
Average access time:	4.0	4.8	5.2	6.0
Random 8KB IOPS rate:	250	208	192	167
146 GB Hitachi FC 10k rpm	25%	50%	75%	100%
Disk rpm:	10,000	10,000	10,000	10,000
Average Write Seek Time:	2.6	3.6	4.1	5.1
Average rotational delay:	3.0	3.0	3.0	3.0
Average access time:	5.6	6.6	7.1	8.1
Random 8KB IOPS rate:	180	152	141	123
750 GB Seagate SATA 7.2k rpm	25%	50%	75%	100%
Disk rpm:	7,200	7,200	7,200	7,200
Average Write Seek Time:	5.0	7.0	8.0	10.0
Average rotational delay:	4.2	4.2	4.2	4.2
Average access time:	9.2	11.2	12.2	14.2
Random 8KB IOPS rate:	109	90	82	71

Table 10. Calculated maximum disk Write IOPS rates by type and active surface usage

The two tables below indicate what IOPS rates (small block random) might be expected from RAID Groups of different types when using 15K RPM SAS disks or 7200 RPM SATA-II disks. The disk IOPS reference values were taken from Tables 9 and 10 above, from the 50% surface usage columns. The left side of each table is the expected maximum rate where the disks are operating at 100% busy (not recommended). The right side of each table shows a more realistic 30% busy rate. For example, when using 15K RPM SAS drives with RAID-5 (7D+1P) at a 30% busy rate, you should plan for about 540 IOPS for 100% Reads, or 125 IOPS for 100% Writes. These are all cache miss values, so the true read rate can increase considerably as much greater cache hit rates are experienced.

Total Disk IOPS by RAID Group (100% busy) 15k disks			Total Disk IOPS by RAID Group (30% busy) 15k disks		
Using IOPS =	*225*	*208*	Using IOPS =	*225*	*208*
RAID Type	IOPS available for Random Read	IOPS available for Random Write	RAID Type	IOPS available for Random Read	IOPS available for Random Write
RAID-10			RAID-10		
2+2	900	416	2+2	270	125
4+4	1,800	832	4+4	540	250
RAID-5			RAID-5		
3+1	900	208	3+1	270	62
7+1	1,800	416	7+1	540	125
RAID-6			RAID-6		
6+2	1,800	277	6+2	540	83

Table 11. Calculated RAID Group IOPS rates (SAS) by RAID level and %busy rates

Total Disk IOPS by RAID Group (100% busy) 7.2k SATA			Total Disk IOPS by RAID Group (30% busy) 7.2k SATA		
Using IOPS =	*99*	*90*	Using IOPS =	*99*	*90*
RAID Type	IOPS available for Random Read	IOPS available for Random Write	RAID Type	IOPS available for Random Read	IOPS available for Random Write
RAID-10			RAID-10		
2+2	396	90	2+2	119	27
4+4	792	180	4+4	238	54
RAID-5			RAID-5		
3+1	396	60	3+1	119	18
7+1	792	120	7+1	238	36
RAID-6			RAID-6		
6+2	792	80	6+2	238	24

Table 12. Calculated RAID Group IOPS rates (SATA-II)by RAID level and %busy rates

Appendix G. AMS500 RAID Group layout example

This section is here in order to illustrate the much more complex configuration considerations when setting up RAID Groups and LUNs on an AMS10000 Family array.

When using an AMS500 with FC disks as an example, if configuring four enclosures (60 disks) for 13 RAID-5 (3D+1P) RAID Groups with four spares, the following layout could have been selected. It looks very clean, as all disks per RAID Group are evenly and vertically distributed among the four enclosures. Some users would like to use this type of layout as it puts each disk per RAID Group in a unique disk enclosure, so that a hardware failure of an enclosure (a very rare occurrence) would only take one disk per RAID Group offline. This same layout could also have been for a RAID-10 2D+2D configuration.

2D+2D or 3D+1P Drive Configuration (AMS500) - BAD METHOD															
Backend	2	3	2	3	2	3	2	3	2	3	2	3	2	3	2
Loop	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
RKA 5 (2,3)															
RKA 4 (0,1)															
RKA 3 (2,3)	R	R	R	R	R	R	R	R	R	R	R	R	R	R	S
RKA 2 (0,1)	G	G	G	G	G	G	G	G	G	G	G	G	G	G	S
RKA 1 (2,3)	0	0	0	0	0	0	0	0	0	0	1	1	1	1	S
RKA 0 (0,1)	0	1	2	3	4	5	6	7	8	9	0	1	2	3	S
HDD Slot:	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

RG of CTL0	RG of CTL1	Spare Drive

Figure 19. Seemingly elegant but very poor RAID Group layout for RAID5 3d+1p on AMS500

However, when analyzing the distribution of disks across the back-end FC loops (four of them), you would see that there is a serious imbalance across the loops, with 28 disks on Loop-0 and none on Loop-1.

CTL:	CTL0		CTL1	
Backend Path:	0	1	0	1
RKA 3	7	0	0	7
RKA 2	7	0	0	7
RKA 1	7	0	0	7
RKA 0	7	0	0	7
Total	28	0	0	28

Figure 20. Disk balance across the loops on AMS500

In order to fix this problem, the default linear allocation method (with some spares added at the end of the enclosures) would have provided a balanced configuration, with 14 disks assigned to each back-end loop.

2D+2D or 3D+1P Drive Configuration (AMS500) - DEFAULT LINEAR METHOD															
Backend	2	3	2	3	2	3	2	3	2	3	2	3	2	3	2
Loop	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
RKA 5 (2,3)															
RKA 4 (0,1)															
RKA 3 (2,3)			RG11				RG12				RG13				S
RKA 2 (0,1)	RG7				RG8				RG9				RG10		S
RKA 1 (2,3)			RG4				RG5				RG6				S
RKA 0 (0,1)	RG0				RG1				RG2				RG3		S
HDD Slot:	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

Figure 21. A good disk allocation method for AMS500

Now, when analyzing the distribution of disks across the back-end FC loops (four of them), you would see that there is a perfect balance across the loops, with 14 disks per loop.

CTL:	CTL0		CTL1	
Backend Path:	0	1	0	1
RKA 3	3	3	4	4
RKA 2	3	3	4	4
RKA 1	4	4	3	3
RKA 0	4	4	3	3
Total	14	14	14	14

Figure 22. Even disk to loop balance

As the RAID levels got larger (i.e. RAID-5 8D+1P, RAID-10 6D+6D), or many RAID levels were mixed together (i.e. RAID-5 3D+1P, RAID-10 3D+3D, RAID-6 5D+2P), it becomes more difficult to make sure the disks per RAID Group and per system are evenly balanced across the available loops.

When using SATA drives in a system, they needed to be laid out in vertical stripes across the enclosures on the same loop, as shown in Figure 2. The DTCL chip logic saw a SATA RAID Group as a single logical disk, and the performance was best when all SATA disks in a RAID Group were on the same loop. But this was very bad for FC disks. So, in configuring an AMS500 or AMS1000 with a mix of SATA and FC disks and different RAID levels, it could be a major task to lay the RAID Groups out in an optimum fashion. This manual layout task continues when new disks are added and new RAID Groups configured.