HP World 2002

Performance Improvement through Effective Disk Allocation and Logical Volume Striping - 258

Danny B. Gross Member of Technical Staff Motorola, Inc 6501 William Cannon West Austin, TX 78735 (512) 895-4825 danny.gross@motorola.com	Chris D. Roberson Storage Solutions Architect Lead Hewlett-Packard 12401 Research Blvd, Suite 200 Austin, TX 78759 (512) 257-5721 chris.roberson@hp.com

Table of Contents

Abstract

This talk provides detailed techniques and instructions for allocating disks within an XP256/512 environment and for utilizing LVM striping to improve volume performance. A case study will provide a unique opportunity to view "before and after" data from a real-life, large-scale data migration in a manufacturing environment. Disk performance analysis techniques using "sar" and PerfView will be reviewed, as will techniques for re-striping logical volumes online.

Architecture of the XP512/XP1024 Disk Array

The internal architecture of the XP512/XP1024 disk array provides the building blocks to design and architect an efficient data layout for optimal performance. The internal architecture of the XP512/XP1024 consists of five major components: Array Control Processors (ACP), Client Host Interface Processors (CHIP), Crossbar, Cache and Shared Memory, and Disk Drives. The components provide the connectivity from the SAN to the individual disk drives.

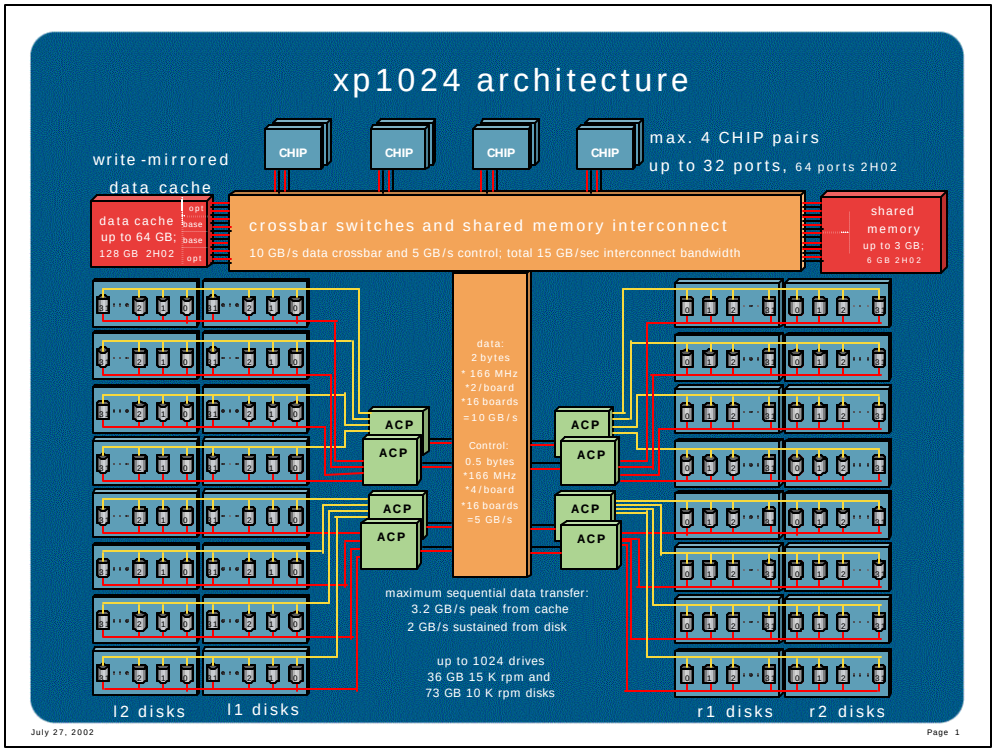


Figure 1 – XP1024 Architecture

Client Host Interface Processor

The Client Host Interface Processor (CHIP) provides the port connectivity to an individual host or storage network. Two varieties of CHIPs exist to provide Fibre Channel or ESCON connectivity. A Fibre Channel CHIP provides connectivity to hosts using UNIX or Microsoft Windows Operating Systems. An ESCON CHIP provides connectivity to mainframe systems using OS390.

CHIPs connect the internal crossbar infrastructure to all hosts. CHIPs utilize redundant circuitry with the ability to reroute data in the event of circuit level failures. The XP Disk Array utilizes a Fibre Channel CHIP in pairs connected to separate power boundaries in the disk array. Each CHIP provides four ports of connectivity with a LUN exposed on identical ports from each card in the CHIP pair. This presentation allows hosts direct or network attached to remain on-line without service interruption whenever upgrading firmware.

Array Control Processor

The Array Control Processor (ACP) provides connectivity to all disk drives in the XP disk array. The ACPs utilize fibre channel technology to attach disk drives to the main system cross bar.

The XP attaches redundant active fibre channel loops (100MB/sec) to dual-ported disk drives. Redundant circuitry with I/O fail-over capability supports rerouting data in the event of circuit level failures. Every ACP provides four fibre channel loops to connect disk drives and a pair of ACPs provides two independent paths to every disk drive. The disk drives support active/active port configurations providing the capability of accepting simultaneous I/O request. The ability to load balance I/O bandwidth under normal operating conditions enables the frame to automatically optimize all internal bandwidth. In cases where path failures occur the operational paths continue to provide access to all data disks while suffering minimal performance degradation.

Crossbar

The XP uses point-to-point and high-speed crossbar interconnects to support attachments throughout the disk array. The point-to-point interconnect supports connectivity between the shared memory to the ACPs and CHIPs. The crossbar interconnect attaches the system cache to the ACPs and CHIPs.

Each ACP and CHIP provides four point-to-point interconnects to the shared memory subsystem. The XP splits shared memory across two power boundaries ensuring system continuity in the event of a power failure. The four interconnects are paired to connect each ACP or CHIP to both sets of shared memory.

Each ACP and CHIP attaches a pair of connections to a redundant high-speed crossbar. The redundant crossbar connects the ACPs and CHIPs to the system cache. A fully configured XP uses four ports to attach to independent cache platform boards. The redundant crossbar utilizes separate power boundaries to ensure availability.

Cache and Shared Memory

The cache subsystem functions as the heart of the entire XP disk array. All read and write operations pass through the cache subsystem. This fact requires the cache subsystem to provide robustness to ensure data integrity under dire circumstances. All system configuration tables are mapped to the shared memory subsystem, segregating system and data traffic.

The cache subsystem of the XP supports duplex writes to multiple cache locations stationed on separate power boundaries. This configuration allows the XP to release hosts immediately after committing a write I/O to cache significantly decreasing host latency. The XP cache architecture allows the coalescing of write I/Os into larger blocks providing increased efficiency when destaging data from cache to disk. The cache battery backup provides support in the event of full power loss to the XP disk array. Fully charged batteries provide 48 hours of on-line backup to restore power to the XP disk array.

The XP uses the shared memory subsystem for managing the system tables. The system tables include configuration data of the XP, logical to physical disk mapping, cache location, and track tables for business continuity. Segregating all system configurations in shared memory allows all cache to be used for data purposes.

Disk Drives

The XP disk array supports high-speed/low-capacity drives for high performance and low-speed/ high-capacity drives for large capacity configurations. A combination of all drive types can be utilized within a

single storage unit allowing compromise between performance and capacity. A four disk group, known as an array group, configured in hardware RAID10 (Mirrored-Striped) or RAID5 (Parity-Striped) provides redundant data protection. A combination of RAID sets can be utilized to support building efficient groupings for intense thorough-put using RAID10 and capacity optimization using RAID5 groupings for cost savings.

The format process slices the physical RAID10/RAID5 array group into virtual disks, known as LDEVs. The LDEV emulation type enforces a consistent size for each LDEV within the array group. The emulation type can vary the size of the LUN from 3GB to 36GBs with 7GB and 14GB being the most frequently chosen size.

Typically an optimum emulation type exists for each drive size used in an XP. Each LDEV appears as an individual physical disk or LUN to the assigned host. Volume management applications, i.e. LVM or VxVM, allow creation of host based meta-volumes. Performance testing indicates host based volume management provides higher throughput than frame based volumes (LUSEs). Choosing a single emulation type for the entire frame decreases management of data location and increases data migration flexibility.

Successful Storage Management and Layout

Any exceptional solution typically simplifies a problem to the greatest degree. Storage utilization on open systems provides a constant challenge for all IT professionals. Many storage industry consultants measure average storage utilization at 45% to 50% of available capacity. Poor utilization not only means everyone pays twice as much for storage on average, but management of that storage enterprise creates headaches for the storage team.

The key to successful enterprise storage array management requires the storage management team view every user of the storage infrastructure with equal importance. By normalizing every storage requirement across the enterprise, a storage administrator quickly identifies available storage resources and optimizes the storage infrastructure automatically. This view of the storage universe means databases, network attached storage, and any other storage space receives the identical priorities and service levels.

The following section details techniques to build consistent and manageable disk processes within your enterprise. These processes utilize software and hardware to efficiently optimize storage capacity and performance. The initial process begins with the identification of storage requirements including capacity, availability, and performance. The high-level requirement definition translates into a detailed specification for all hosts connected to the storage infrastructure. After finalizing the host layout, a disk array layout definition optimally locates all data within the enterprise storage disk array.

Storage Requirements

The storage requirement definition provides a roadmap to identify the proper classification and architecture for the entire storage infrastructure. The typical storage requirement definition begins with identifying all application storage needs. The application needs include availability, capacity, and performance.

With the complexity of today's storage networks, you may find the application storage needs vary widely from intense on-line transaction processing to simple data repository catalogs. A requirements map built to provide the application, host operating system, performance requirements, and capacity assists with depicting the storage architecture.

Table-1 illustrates an example environment supporting an enterprise resource planning system, i.e. SAP, and a data warehouse. Each environment maintained substantially different availability, performance and capacity requirements. The differences caused the architecture team to create two separate storage environments to

Equiva Storage Requirements					
Application	**Hosts**	**Availability**	**Performance**	**Capacity**	**Notes**
SAP					
	mapdb00p	Clustered	Random	4100	
	mapdb02p	Clustered	Random	0	
	mapin00p		Random	400	
	ndceqbk1		Sequential	4100	
Data Warehouse					
	mawdb01p		Sequential	1800	
	lumdb05p		Sequential	500	
	tstrept		Sequential	500	
	ndceqbk2		Sequential	1800	

Table 1 – Storage Requirements Map

optimize the infrastructure for the specific application.

The storage requirements map becomes the definitive guide for designing the host layout maps. The basic requirements definition assists with determining the enterprise storage infrastructure and the suitability of the architecture to the application landscape.

Data Layout Definition

A data layout definition details the configuration for each host connected to the storage network. The definition provides the implementation plan for all hosts attached to a storage network or network-attached storage. The definition includes host layout information like: mount points, volume management information, and disk device requirements.

A storage architect evaluates two types of environments when creating a data layout definition. A new install or "Greenfield" environment provides an opportunity to create a new plan from scratch. In most cases a preexisting environment must be retrofitted to the new storage infrastructure.

A Greenfield environment's most challenging problem surrounds identifying the precise capacity for each host application. The storage architect must interact with two teams to discover the configuration needed to implement the storage infrastructure. The application team provides detail on capacity, performance, and application

characteristics. The infrastructure team identifies the availability, business continuity, host connectivity, and host layout conventions, i.e. mount point and volume group nomenclature.

Force Production Existing Layout				Force Production Migrating Layout					
VG	LV	Mount Point	Requested LVM Total (Mbytes)	VG	LV	Mount Point	Open-E LDEVs	VG Total (Mbytes)	LVM Total (Mbytes)
vg03	lvol1	/u003	5000	vg001	lvol1	/u003	4	28112	7028
vg09	lvol1	/u009	2000		lvol2	/u009			7028
vg04	lvol1	/u004	5000		lvol3	/u004			7028
vg02	lvol1	/u002	7000	vg002	lvol1	/u002	4	28112	7028
vg10	lvol1	/u010	2000		lvol2	/u010			7028
vg12	lvol1	/u012	5000		lvol3	/u012			7028
vg01	lvol1	/u001	18000	vg003	lvol1	/u001	12	84336	28112
vg05	lvol1	/u005	9000		lvol2	/u005			28112
vg08	lvol1	/u008	8000		lvol3	/u008			28112
vg07	lvol1	/u007	6000	vg004	lvol1	/u007	12	84336	28112
vg06	lvol1	/u006	7000		lvol2	/u006			28112
vg11	lvol1	/u011	12000		lvol3	/u011			28112
Total:			**86000**						**210840**

Force Non-Production Existing Layout				Force Non-Production Migrating Layout					
VG	LV	Mount Point	Requested LVM Total (Mbytes)	VG	LV	Mount Point	Open-E LDEVs	VG Total (Mbytes)	LVM Total (Mbytes)
vg09	lvol1	/u109	6000	vg101		/u109	4	28112	7028
vg03	lvol1	/u103	15000			/u103			14056
vg04	lvol1	/u104	15000	vg102		/u104	4	28112	14056
vg12	lvol1	/u112	15000			/u112			14056
vg02	lvol1	/u102	21000	vg103		/u102	4	28112	21084
vg10	lvol1	/u110	6000			/u110			7028
vg05	lvol1	/u105	27000	vg104		/u105	12	84336	42168
vg08	lvol1	/u108	24000			/u108			42168
vg07	lvol1	/u107	18000	vg105		/u107	12	84336	42168
vg06	lvol1	/u106	21000			/u106			42168
vg11	lvol1	/u111	36000	vg106		/u111	12	84336	28112
vg01	lvol1	/u101	54000			/u101			56224
Total:			**258000**						**330316**

Yavin Production Existing Layout				Yavin Production Migrating Layout					
VG	LV	Mount Point	Requested LVM Total (Mbytes)	VG	LV	Mount Point	Open-E LDEVs	VG Total (Mbytes)	LVM Total (Mbytes)
vg01	lvol1	/u201	13	vg200	lvol1	/u201	3	21084	21084
vg02	lvol1	/u201	13	vg201	lvol1	/u202	6	42168	42168
Total:			**26**						**63252**

Endor Production Existing Layout				Endor Production Migrating Layout					
VG	LV	Mount Point	Requested LVM Total (Mbytes)	VG	LV	Mount Point	Open-E LDEVs	VG Total (Mbytes)	LVM Total (Mbytes)
vg01	lvol1	/u201	13	vg300	lvol1	/u301	3	21084	21084
vg02	lvol1	/u201	13	vg301	lvol1	/u302	6	42168	42168
Total:			**26**						**63252**

Table 2 – "Greenfield" Disk Layout Definition

HP storage architects recently deployed a "Greenfield" Oracle Financial environment. With the assistance of the application and infrastructure team, the storage architect created the data layout definition in Table-2. The definition identified the key applications, forms and the oracle database. A data definition for each host provided the details for configuring the volume management and file system structure for each host.

A redeployed environment's largest challenge requires inventorying the existing environment and defining a method to retrofit it into the new storage infrastructure. The storage architect must collect information from all hosts identifying the volume management, connectivity, and capacity requirements. The inventory provides the needed information to detail the data layout definition. The architect needs to collaborate with the infrastructure team to modify the existing layout inventory to a new data definition. The infrastructure team identifies acceptable nomenclature for the volume management and file system layout.

A recently retrofitted environment redeployed enterprise storage from EMC to a new XP512 based infrastructure. Table-3 details the inventoried database server configuration and the redeployed configuration used to support the XP512 storage. The definition identified the capacity requirement of the existing environment and provided additional storage for anticipated growth.

Existing Layout							Migrating Layout						
VG	LV	Mount Point	BDF Total (Mbytes)	BDF Used (Mbytes)	BDF Free (Mbytes)	LVM Total (Mbytes)	VG	LV	Mount Point	Open-E LDEVs	VG Total (Mbytes)	LVM Total (Mbytes)	Datafile Migration (Mbytes)
vg_ixos	itadm	/itadm	4096	3590	474	4000	vg_ixos	itadm	/itadm	4	55488	5000	
vg_ixos	itparch	/archive/ITP	5128	102	4712	5008		itparch	/archive/ITP			5000	
vg_ixos	itporadata1	/oradata1/ITP	5128	3202	1806	5008		itporadata1	/oradata1/ITP			5000	
vg_ixos	itporadata2	/oradata2/ITP	5128	2587	2382	5008		itporadata2	/oradata2/ITP			5000	
vg_ixos	itporadata3	/oradata3/ITP	5128	2613	2358	5008		itporadata3	/oradata3/ITP			5000	
vg_ixos	itporadata4	/oradata4/ITP	3080	2649	404	3008		itporadata4	/oradata4/ITP			5000	
vg_ixos	taxfirm	/export/prddata/multi/taxfirm	20480	4	20157	20000		taxfirm	/export/prddata/multi/taxfirm			20000	
vg_nfs_infr	lv_controlm	/opt/controlm	2572	2115	429	2512	vg_nfs_infr	lv_controlm	/opt/controlm	4	55488	2500	
vg_nfs_infr	lv_ctmagent	/opt/ctmagent	262	79	174	256		lv_ctmagent	/opt/ctmagent			2500	
vg_nfs_infr	lv_ecs	/opt/ecs	2048	1070	917	2000		lv_ecs	/opt/ecs			2500	
vg_nfs_infr	lv_perfdata	/export/perfdata	3080	2	2886	3008		lv_perfdata	/export/perfdata			5000	
vg_nfs_infr	lv_prddata	/export/prddata	22561	19606	2909	22032		lv_prddata	/export/prddata			25000	
vg_nfs_infr	lv_prddata_datawh	/export/prddata/multi/datawh/arch	10240	6854	3280	10000		lv_prddata_datawh	/export/prddata/multi/datawh/arch			10000	
vg_nfs_infr	lv_prod_doc	/export/jobsched	49	8	35	48		lv_prod_doc	/export/jobsched			2500	
vg_nfs_infr	lv_scripts	/export/scripts	164	73	83	160		lv_scripts	/export/scripts			2500	
vg_oracle_infr	lv_apps	/export/mapin01p	9011	7890	1088	8800	vg_oracle_infr	lv_apps	/export/mapin01p	4	55488	10000	
vg_oracle_infr	lv_arch	/export/mapin01p1/arch	1032	1	966	1008		lv_arch	/export/mapin01p1/arch			2500	
vg_oracle_infr	lv_controlm_bk	/controlm_backup	4096	561	3314	4000		lv_controlm_bk	/controlm_backup			5000	
vg_oracle_infr	lv_oracle	/oracleINFR	410	1	383	400		lv_oracle	/oracleINFR			2500	
vg_oracle_infr	lv_oracle_8	/oracle	1540	6	1438	1504		lv_oracle_8	/oracle			2500	
vg_oracle_infr	lv_oracle1	/export/mapin01p1	1573	533	975	1536		lv_oracle1	/export/mapin01p1			2500	
vg_oracle_infr	lv_oracle2	/export/mapin01p2	1573	416	1085	1536		lv_oracle2	/export/mapin01p2			2500	
vg_oracle_infr	lv_oracle3	/export/mapin01p3	1573	1235	317	1536		lv_oracle3	/export/mapin01p3			2500	

Table 3 – Retrofitted Disk Data Definition

The data layout definition provides the details needed to implement a host into the storage infrastructure. The infrastructure team utilizes the layout definition to construct a layout for each host and connect the host to the storage infrastructure.

Logical Disk Layout

A logical disk layout identifies the configuration of the storage disk array connected to the storage network. The layout provides the implementation plan for all storage attached to the storage network or network-attached storage. It further depicts the mapping of the

4-8	4-7	4-6	4-5	4-4	4-3	4-2	4-1			2-1	2-2	2-3	2-4	2-5	2-6	2-7	2-8
CU:LDEV	CU:LDEV	CU:LDEV	CU:LDEV	CU:LDEV	CU:LDEV	CU:LDEV	CU:LDEV			CU:LDEV	CU:LDEV	CU:LDEV	CU:LDEV	CU:LDEV	CU:LDEV	CU:LDEV	CU:LDEV
3:69	3:5A	3:4B	3:3C	3:2D	3:1E	3:0F	3:00			1:00	1:0F	1:1E	1:2D	1:3C	1:4B	1:5A	1:69
3:6A	3:5B	3:4C	3:3D	3:2E	3:1F	3:10	3:01			1:01	1:10	1:1F	1:2E	1:3D	1:4C	1:5B	1:6A
3:6B	3:5C	3:4D	3:3E	3:2F	3:20	3:11	3:03			1:02	1:11	1:20	1:2F	1:3E	1:4D	1:5C	1:6B
3:6C	3:5D	3:4E	3:3F	3:30	3:21	3:13	3:03			1:03	1:12	1:21	1:30	1:3F	1:4E	1:5D	1:6C
3:6D	3:5E	3:4F	3:40	3:31	3:22	3:13	3:04			1:04	1:13	1:22	1:31	1:40	1:4F	1:5E	1:6D
3:6E	3:5F	3:50	3:41	3:32	3:23	3:14	3:05			1:05	1:14	1:23	1:32	1:41	1:50	1:5F	1:6E
3:6F	3:60	3:51	3:43	3:33	3:24	3:15	3:06			1:06	1:15	1:24	1:33	1:42	1:51	1:60	1:6F
3:70	3:61	3:53	3:43	3:34	3:25	3:16	3:07			1:07	1:16	1:25	1:34	1:43	1:52	1:61	1:70
3:71	3:62	3:53	3:44	3:35	3:26	3:17	3:08			1:08	1:17	1:26	1:35	1:44	1:53	1:62	1:71
3:72	3:63	3:54	3:45	3:36	3:27	3:18	3:09			1:09	1:18	1:27	1:36	1:45	1:54	1:63	1:72
3:73	3:64	3:55	3:46	3:37	3:28	3:19	3:0A			1:0A	1:19	1:28	1:37	1:46	1:55	1:64	1:73
3:74	3:65	3:56	3:47	3:38	3:29	3:1A	3:0B			1:0B	1:1A	1:29	1:38	1:47	1:56	1:65	1:74
3:75	3:66	3:57	3:48	3:39	3:2A	3:1B	3:0C			1:0C	1:1B	1:2A	1:39	1:48	1:57	1:66	1:75
3:76	3:67	3:58	3:49	3:3A	3:2B	3:1C	3:0D			1:0D	1:1C	1:2B	1:3A	1:49	1:58	1:67	1:76
3:77	3:68	3:59	3:4A	3:3B	3:2C	3:1D	3:0E			1:0E	1:1D	1:2C	1:3B	1:4A	1:59	1:68	1:77

3-8	3-7	3-6	3-5	3-4	3-3	3-2	3-1			1-1	1-2	1-3	1-4	1-5	1-6	1-7	1-8
CU:LDEV	CU:LDEV	CU:LDEV	CU:LDEV	CU:LDEV	CU:LDEV	CU:LDEV	CU:LDEV			CU:LDEV	CU:LDEV	CU:LDEV	CU:LDEV	CU:LDEV	CU:LDEV	CU:LDEV	CU:LDEV
2:69	2:5A	2:4B	2:3C	2:2D	2:1E	2:0F	2:00			0:00	0:0F	0:1E	0:2D	0:3C	0:4B	0:5A	0:69
2:6A	2:5B	2:4C	2:3D	2:2E	2:1F	2:10	2:01			0:01	0:10	0:1F	0:2E	0:3D	0:4C	0:5B	0:6A
2:6B	2:5C	2:4D	2:3E	2:2F	2:20	2:11	2:02			0:02	0:11	0:20	0:2F	0:3E	0:4D	0:5C	0:6B
2:6C	2:5D	2:4E	2:3F	2:30	2:21	2:12	2:03			0:03	0:12	0:21	0:30	0:3F	0:4E	0:5D	0:6C
2:6D	2:5E	2:4F	2:40	2:31	2:22	2:13	2:04			0:04	0:13	0:22	0:31	0:40	0:4F	0:5E	0:6D
2:6E	2:5F	2:50	2:41	2:32	2:23	2:14	2:05			0:05	0:14	0:23	0:32	0:41	0:50	0:5F	0:6E
2:6F	2:60	2:51	2:42	2:33	2:24	2:15	2:06			0:06	0:15	0:24	0:33	0:42	0:51	0:60	0:6F
2:70	2:61	2:52	2:43	2:34	2:25	2:16	2:07			0:07	0:16	0:25	0:34	0:43	0:52	0:61	0:70
2:71	2:62	2:53	2:44	2:35	2:26	2:17	2:08			0:08	0:17	0:26	0:35	0:44	0:53	0:62	0:71
2:72	2:63	2:54	2:45	2:36	2:27	2:18	2:09			0:09	0:18	0:27	0:36	0:45	0:54	0:63	0:72
2:73	2:64	2:55	2:46	2:37	2:28	2:19	2:0A			0:0A	0:19	0:28	0:37	0:46	0:55	0:64	0:73
2:74	2:65	2:56	2:47	2:38	2:29	2:1A	2:0B			0:0B	0:1A	0:29	0:38	0:47	0:56	0:65	0:74
2:75	2:66	2:57	2:48	2:39	2:2A	2:1B	2:0C			0:0C	0:1B	0:2A	0:39	0:48	0:57	0:66	0:75
2:76	2:67	2:58	2:49	2:3A	2:2B	2:1C	2:0D			0:0D	0:1C	0:2B	0:3A	0:49	0:58	0:67	0:76
2:77	2:68	2:59	2:4A	2:3B	2:2C	2:1D	2:0E			0:0E	0:1D	0:2C	0:3B	0:4A	0:59	0:68	0:77

vgdb01	vgdb37	vgdb19	
vgdb02	vgdb38	vgdb20	
vgdb03	vgdb39	vgdb21	
vgdb04	vgdb40	vgdb22	
vgdb05	vgdb41	vgdb23	
vgdb06	vgdb42	vgdb24	
vgdb07		vgdb25	
vgdb08		vgdb26	
vgdb09		vgdb27	
vgdb10		vgdb28	
vgdb11		vgdb29	
vgdb12		vgdb30	
vgdb13		vgdb31	
vgdb14		vgdb32	
vgdb15	vgsap01	vgdb33	
vgdb16	vgsap02	vgdb34	
vgdb17	vgsap03	vgdb35	SAP Business Copy

Table 4 – Logical Disk Layout Map

volume group structure of each host on the storage array.

A storage architect converts the data layout definition into an optimal disk layout. An optimal disk layout utilizes the storage array infrastructure to provide a long-term manageable and scalable environment. To define the optimal layout a rule set applied to the definition yields the best overall structure by trading-off capacity and performance.

A typical rule set used for an HP StorageWorks XP Disk Array looks like the following:

- Volume groups can be identified and imported on any host. Each disk on the array has path redundancy through its physical volume link.
- Logical volumes are treated identically, independent of application.
- Array features are fully exploited for every logical volume.
- The RAID5 write penalty is minimized across a RAID0 stripe across XP ACP's.

To achieve a better understanding of applying the rule set listed above, consider the following example:

Table-4 illustrates converting the data layout definition from the retrofitted layout example (Table-3). The layout utilizes color-coding to quickly visualize the volume group structure across the entire disk array. The layout map also provides the LUN/Port mappings of the disk array to support connecting the disk array to the storage network.

The steps taken to allocate volumes in the LUN Map are as follows:

1. From the Data Definition Layout (Table-3), determine the number of physical volumes required for the volume group, i.e., vgDB19 requires four LUNs.
2. Locate and color code an unallocated CU:LDEV in the LUN Map. Using our example for vgDB19, we selected LUN 0:0B (ACP 0, bottom right quadrant).

3. From here, allocate the same LDEV in the remaining ACP quadrants. For vgDB19, we highlighted LDEVs 0:1B,1:1B, 2:1B, 3:1B.
4. When these disks are extended into a volume group, insure that they are identified with the same physical volume group, for striping.

Using these steps, we have automatically and rapidly allocated LUN's while meeting the rules for maximum performance and utilization. If we allocate all volume groups on the array in the same manner, we can be assured that every physical volume group will perform in exactly the same way. It will no longer be possible for a "hot disk" to exist.

The Data Definition Layout provides the planning data for LUN allocation on the XP storage framework. The LUN map is a tool that assists in allocating LUNs within the XP framework for maximum effectiveness and performance. This disk layout provides the details to implement the disk array into the storage infrastructure.

Implementing the data layout definition

Once the logical disk layout has been planned, the storage architect reviews the operational requirements of the infrastructure to strategize its implementation. There are several methods that can be used in combination to execute the required changes:

Use the "Greenfield" approach to implement the logical disk layout

1. Create fully new disk infrastructure, and copy or restore data to the replacement disks.
2. When the restore is complete, dismount the source storage, and mount the replacement storage in place.

This approach has the benefit of data safety. For a very small amount of change, or in the case where the storage architect desires to repair damaged filesystems, this method is very effective. However, for a large-scale implementation, this method can be tremendously costly in terms of systems and applications outage and complexity.

Relocate logical volume extents to target locations on a disk-for-disk basis

1. Use the command "pvmove" to relocate the extents of a logical volume from one disk to another. "pvmove <disk1> <disk2>" will

relocate all the extents from disk 1 to disk2. "pvmove –n <logical volume> <disk1> <disk2>" relocates all of the extents of <logical volume> that exist on <disk1> to <disk2>.

2. When the relocation is complete, recover the obsolete disks by reducing from the volume group with "vgreduce <VG Name> <disk1> <disk2> …"

This approach is a very safe, effective online means of "surgically" relocating logical extents to a target disk. It works extremely well for a medium amount of change, or to make way in a new location for incoming data. While it can be conducted online with extremely low performance impact (~5% CPU) to the system, this method can be time-consuming. For a 7GB disk with four logical volumes, this command can take upwards of 30 minutes to complete.

Re-stripe logical volume extents by creating mirrors in the target location

1. Organize the target disks into "physical volume groups" by editing /etc/lvmpvg, or by extending disks into the volume group with the "vgextend –g <PVG Name> <Volume Group> <disk1> <disk2> …"
2. Insure that the logical volume is set to distributed and striped. Correct if necessary by using "lvchange –D y –s g <logical volume>"
3. Create a striped mirror into the target physical volume group by using "lvextend –m 1 <logical volume> <PVG Name>. This will create a mirror by placing one extent at a time across every disk in the physical volume group.
4. When complete, reduce the mirror from the obsolete location by using "lvreduce –m 0 <disk1> <disk2> …" where "<disk1> <disk2> …" is a list of disks from which to remove the mirror.
5. Remove the obsolete disks from the volume group by using "vgreduce <VG Name> <disk1> <disk2> …"

This approach is a very safe, effective online means of striping and/or relocating logical volumes to new locations. It is a very good means of restriping large volumes, and of conducting large-scale implementations. The average time to build a striped mirror is approximately 15 minutes per 7GB of logical volume extents.

Measuring the performance of a data layout definition

The most challenging aspect of disk relocation can be not how to conduct it online, but rather how to measure whether it made any difference. There are three tremendous tools that can be used to monitor disk performance and gauge effectiveness of the disk allocation:

The Unix I/O statistics reporter "iostat"

"iostat –t 5" provides basic summary disk performance data for a five second interval in time. While very basic, this command is an excellent gauge for "at-the-moment" disk usage. Sometimes, iostat can provide a very clear view of the effectiveness of the disk allocation. Consider the following example output in Table 5 for a five second interval on an N4000 server:

device	bps	sps	msps
c1t6d0	127	28.5	1.0
c2t6d0	118	24.5	1.0
c8t9d5	10252	44.4	1.0
c6t9d5	125	4.8	1.0
c5t9d5	125	4.8	1.0
c8t9d6	136	5.2	1.0
c6t9d6	2928	210.4	1.0
c5t9d6	362	31.9	1.0
c8t10d4	2792	304.6	1.0
c6t10d4	11035	146.0	1.0
c5t10d4	135	6.0	1.0
c33t1d6	948	8.0	1.0
c35t4d1	948	8.0	1.0
c29t6d0	948	8.0	1.0
c31t10d2	948	8.0	1.0
c26t1d6	948	8.0	1.0
c33t6d0	948	8.0	1.0
c35t8d3	948	8.0	1.0
c29t10d2	948	8.0	1.0
c31t1d6	948	8.0	1.0
c26t6d0	948	8.0	1.0
c33t10d2	948	8.0	1.0
c35t12d5	1205	9.0	1.0

Table 5 – "iostat" Summary

This one five-second snapshot provides striking evidence supporting the concepts behind the techniques described earlier in this paper. This data was from an N4000 Informix Database server, taken with both the

"obsolete" and target layout disks still joined by logical volume mirroring. Below, the data in Table 6 will be highlighted for clarity:

device	bps	sps	msps	
c1t6d0	127	28.5	1.0	vg00
c2t6d0	118	24.5	1.0	vg00
c8t9d5	10252	44.4	1.0	/dev/vg_infmx/lvol1
c6t9d5	125	4.8	1.0	/dev/vg_infmx/lvol1
c5t9d5	125	4.8	1.0	/dev/vg_infmx/lvol1
c8t9d6	136	5.2	1.0	/dev/vg_infmx/lvol1
c6t9d6	2928	210.4	1.0	/dev/vg_infmx/lvol1
c5t9d6	362	31.9	1.0	/dev/vg_infmx/lvol2
c8t10d4	2792	304.6	1.0	/dev/vg_infmx/lvol2
c6t10d4	11035	146.0	1.0	/dev/vg_infmx/lvol2
c5t10d4	135	6.0	1.0	/dev/vg_infmx/lvol2
c33t1d6	948	8.0	1.0	/dev/vg_infmx/lvol2
c35t4d1	948	8.0	1.0	
c29t6d0	948	8.0	1.0	
c31t10d2	948	8.0	1.0	
c26t1d6	948	8.0	1.0	
c33t6d0	948	8.0	1.0	
c35t8d3	948	8.0	1.0	
c29t10d2	948	8.0	1.0	
c31t1d6	948	8.0	1.0	
c26t6d0	948	8.0	1.0	
c33t10d2	948	8.0	1.0	
c35t12d5	1205	9.0	1.0	

Table 6 – "iostat" Summary Analysis

- /dev/vg_infmx/lvol1 is a 35GB filesystem, allocated as shown in green.
- /dev/vg_infmx/lvol2 is a 28GB filesystem, allocated as shown in yellow.
- The mirror for both logical volumes was striped across the 12-disk physical volume group as shown in blue.

Looking at the output, it is fairly easy to see the performance issues with this system:

- /dev/vg_infmx/lvol1 is a distributed but unstriped volume (Extents allocated to fill the first disk, followed by each disk in order). This is evidenced by the fact that the greater amount of I/O is being served by only two of the seven disks. The first disk in the volume is near 100% utilized, evidenced by the very high byte rate.

- /dev/vg_infmx/lvol2 is a similar volume, built on four disks. Its first disk is likewise near 100% utilized, evidenced by the very high byte rate.
- Note the high amount of seeks per second reported by the busier disks in both logical volumes.
- The I/O is very evenly distributed across all twelve of the mirror disks. Note that the number of seeks per second is dramatically lower than the primary side of the mirror.

Prior to the re-allocation of the disks assigned to this particular server, the applications manager reported very bad database performance. Using the data above, we correctly predicted a nearly ten-fold improvement in throughput when the mirrors were split.

The Unix Systems Activity Reporter "sar"

"sar –d 10" provides more detailed summary disk performance data for a ten second interval in time. It provides much more granular data than "iostat", and is likewise limited by the timeliness of its report. However, sar can be used to summarize data over a long period of time for general trend analysis.

Consider the output in Table 7 of a ten second interval for ten intervals sar summary (sar –d 10 10):

system1	B.11.00	A	9000/800	2/4/2002		
device	%busy	avque	r+w/s	blks/s	avwait	avserv
c1t6d0	5.64	0.56	9	71	4.95	9.62
c2t6d0	4.23	0.56	8	65	4.95	7.7
c17t10d7	35.67	0.59	223	2301	5.23	2.12
c17t2d7	27.59	1.9	70	3164	7.49	13
c18t2d7	23.88	1.85	56	3066	7.53	14.2
c10t2d7	22.43	1.75	51	3046	7.31	13.8
c10t10d7	6.47	1.24	35	1804	6.77	5.15
c18t10d7	6.26	0.94	32	1420	5.95	4.01
c10t5d4	0.23	0.5	1	10	3.8	4.32
c31t1d1	22.45	0.52	211	1680	5.08	1.27
c33t1d1	15.41	0.97	53	2226	5.95	8.17
c37t12d0	14.42	1.93	35	2443	7.44	12.89
c29t5d3	13.2	1.38	31	2236	6.53	13.08
c35t9d5	12.88	1.41	32	2213	6.77	12.3
c35t11d4	4.05	1.33	18	1317	8.25	7.71
c29t7d2	3.66	0.86	19	1541	5.73	4.98
c33t3d0	3.05	0.83	16	855	5.92	4.62
c33t4d7	0.05	0.5	0	5	3.34	3.9

Table 7 – "sar" Summary

Like the iostat example above, this data was captured while the "obsolete" and target disks were mirrored together (logical volumes restriped on target). The "obsolete" disks are highlighted in yellow, while the target disks are in blue.

Looking at the data, it is difficult to see a striking difference between the old and new allocations. However, we can roughly analyze the data and approximate the effectiveness of the new disk allocation. The results are actually impressive. First, some definitions:

- %busy – The percentage of time the disk was kept busy servicing requests.
- aveque – The average number of pending disk requests.
- r+w/s – The average number of read/write requests per second.
- blks/s – The average number of 512-byte blocks throughput per second.
- avwait – The average time (in milliseconds) that a request waited on queue
- avserv – The average service time (in milliseconds) for the disk.

Some of the areas in which we would hope to see an improvement as a result of an effective re-allocation are as follows:

- Reduction in average %busy. (data spread evenly across disks)
- Reduction in average r+w/s. (seeks per second spread evenly)
- Reduction in average total wait time for a disk (avewait+aveserv).

Table 8 derives this data from our example:

HP-UX	system1	B.11.00	A	9000/800	2/4/2002							
								ms	ms	ms		
9:39:26	device	%busy	avque	r+w/s	blks/s	avwait	avserv	Totwait	AveTotal wait	Aver+w/s	Aveblks/s	AveBusy
Average	c1t6d0	5.64	0.56	9	71	4.95	9.62	14.57				
Average	c2t6d0	4.23	0.56	8	65	4.95	7.7	12.65				
Average	c17t10d7	35.67	0.59	223	2301	5.23	2.12	7.35				
Average	c17t2d7	27.59	1.9	70	3164	7.49	13	20.49				
Average	c18t2d7	23.88	1.85	56	3066	7.53	14.2	21.73				
Average	c10t2d7	22.43	1.75	51	3046	7.31	13.8	21.11				
Average	c10t10d7	6.47	1.24	35	1804	6.77	5.15	11.92				
Average	c18t10d7	6.26	0.94	32	1420	5.95	4.01	9.96				
Average	c10t5d4	0.23	0.5	1	10	3.8	4.32	8.12	14.38	66.86	2115.86	17.50
Average	c31t1d1	22.45	0.52	211	1680	5.08	1.27	6.35				
Average	c33t1d1	15.41	0.97	53	2226	5.95	8.17	14.12				
Average	c37t12d0	14.42	1.93	35	2443	7.44	12.89	20.33				
Average	c29t5d3	13.2	1.38	31	2236	6.53	13.08	19.61				
Average	c35t9d5	12.88	1.41	32	2213	6.77	12.3	19.07				
Average	c35t11d4	4.05	1.33	18	1317	8.25	7.71	15.96				
Average	c29t7d2	3.66	0.86	19	1541	5.73	4.98	10.71				
Average	c33t3d0	3.05	0.83	16	855	5.92	4.62	10.54				
Average	c33t4d7	0.05	0.5	0	5	3.34	3.9	7.24	13.77	46.11	1612.89	9.91
							Summary:		4.26%	31.03%	23.77%	43.40%

Table 8 – "sar" Summary Analysis

We added the columns:

"Totwait" = "avwait" + "avserv"
"AveTotalwait" = average of "Totwait" for an allocation
"Aver+w/s" = average of "r+w/s" for an allocation
"Aveblks/s" = Average of "blks/s" for an allocation
"AveBusy" = average of "%busy" for an allocation

The summary line provides a very rough, positive approximation of the effectiveness of our new allocation:

- %busy is reduced by 43.4% on average.
- r+w/s is reduced by 31.0% on average.
- Total Wait time is reduced by 4.3% on average.
- Average blks/s is reduced by 23.8% on average.

Sar's rough guess provided validation of our new allocation. As far as this particular system goes, the original logical volumes were distributed and striped. However, half of the original disks were allocated from one ACP pair on two array groups, while the other half were allocated from another ACP pair on two other array groups. The new allocation was fully distributed across four Fibre Channels, 4 ACP pairs, and 8 array groups. Our prediction shows a positive difference using the allocation techniques described earlier. We have definitely reduced the average load on a disk. How good was this approximation? When we split and removed the "obsolete" side of the mirrors on this system, the customer called and told us that something had happened – the application was responding noticeably faster.

Hewlett Packard's PerfView Analyzer

PerfView Analyzer is an outstanding product that allows rapid analysis of hundreds of performance parameters captured by the MeasureWare system. It is capable of summarizing data down to five-minute intervals for several month periods. PerfView Analyzer is a very powerful tool for trend analysis. While iostat and sar provide very good point-in-time data summarizations to gauge effectiveness of

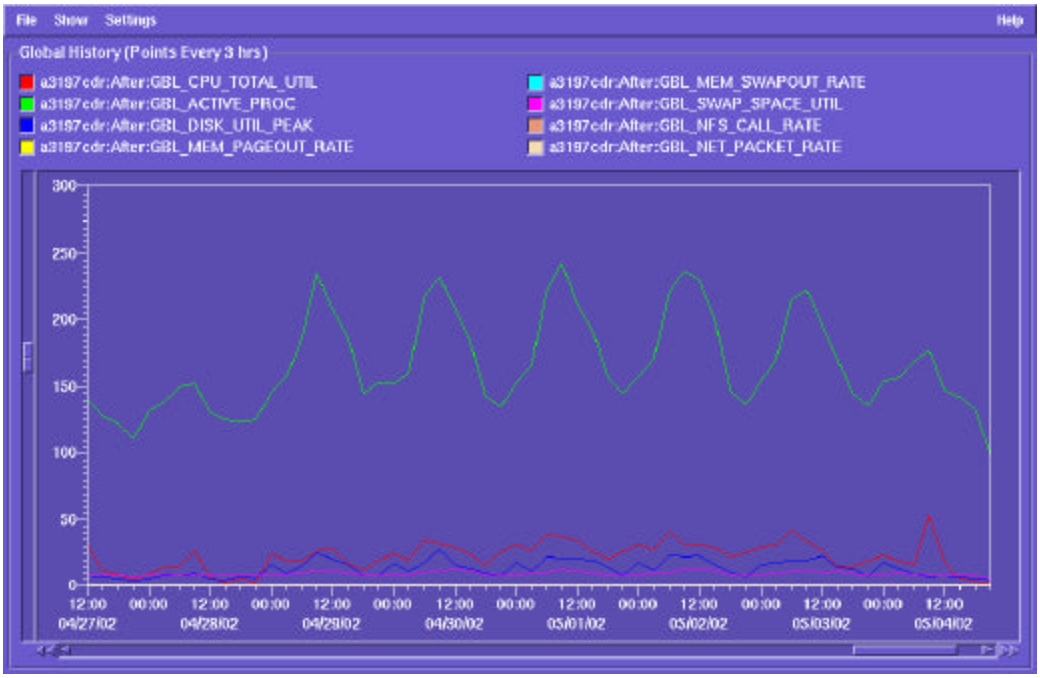
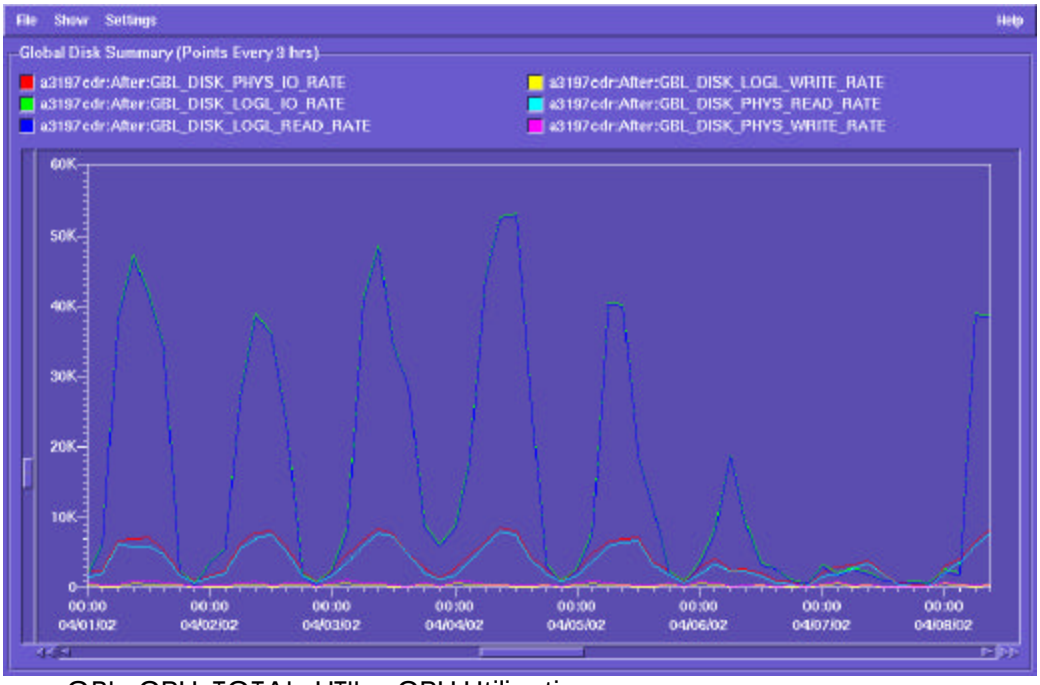


Figure 2 – Global History

allocations, PerfView Analyzer can provide evidence of its impact on system performance as a whole.

The Global History Chart (Figure 2) shows global system performance metrics. Of particular importance are the following:

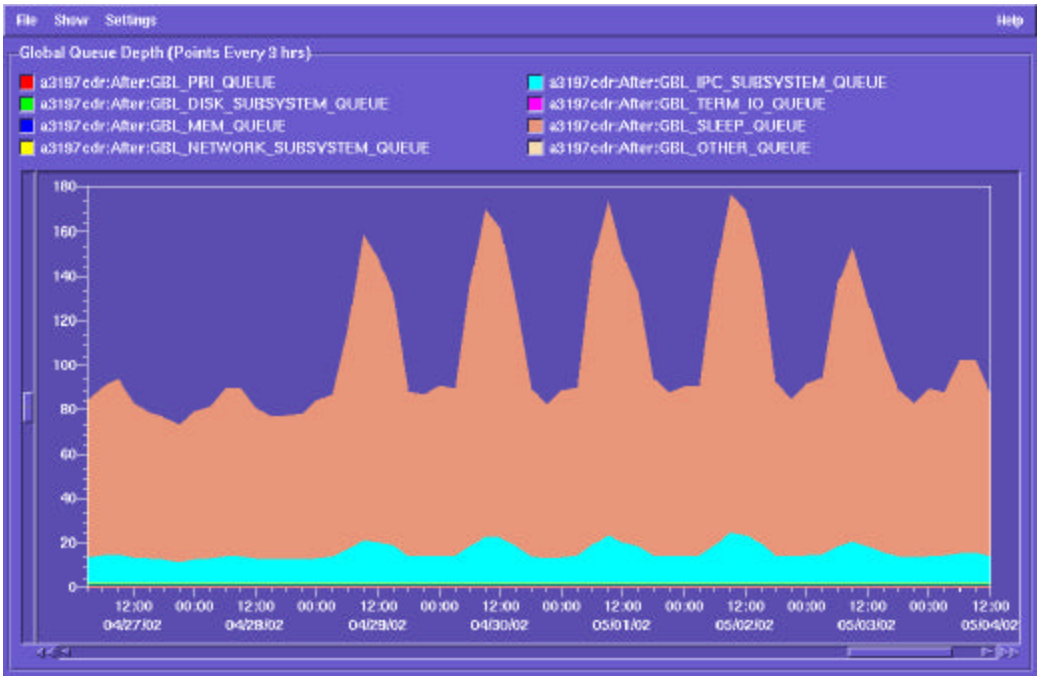


GBL_CPU_TOTAL_UTIL – CPU Utilization
GBL_DISK_UTIL_PEAK –Highest utilization reported for a disk

This Global Disk Summary Chart (Figure 3) shows global disk performance metrics. Of particular importance are the following metrics:

GBL_DISK_UTIL_PEAK - The highest utilization percentage reported by a disk
GBL_DISK_PHYS_IO_RATE - The total physical I/O rate for the system.
GBL_DISK_PHYS_READ_RATE - The total physical read request rate
GBL_DISK_PHYS_WRITE_RATE - The total physical write request rate

Figure 3 – Global Disk Summary



The Global Queue Depth Chart (Figure 4) displays queue information of the system. Of particular importance are the following:

GBL_PRI_QUEUE – The size of the CPU priority queue
GBL_DISK_SUBSYSTEM_QUEUE – The size of the disk request queue.
GBL_IPC_SUBSYSTEM_QUEUE – the size of the interprocess communications queue

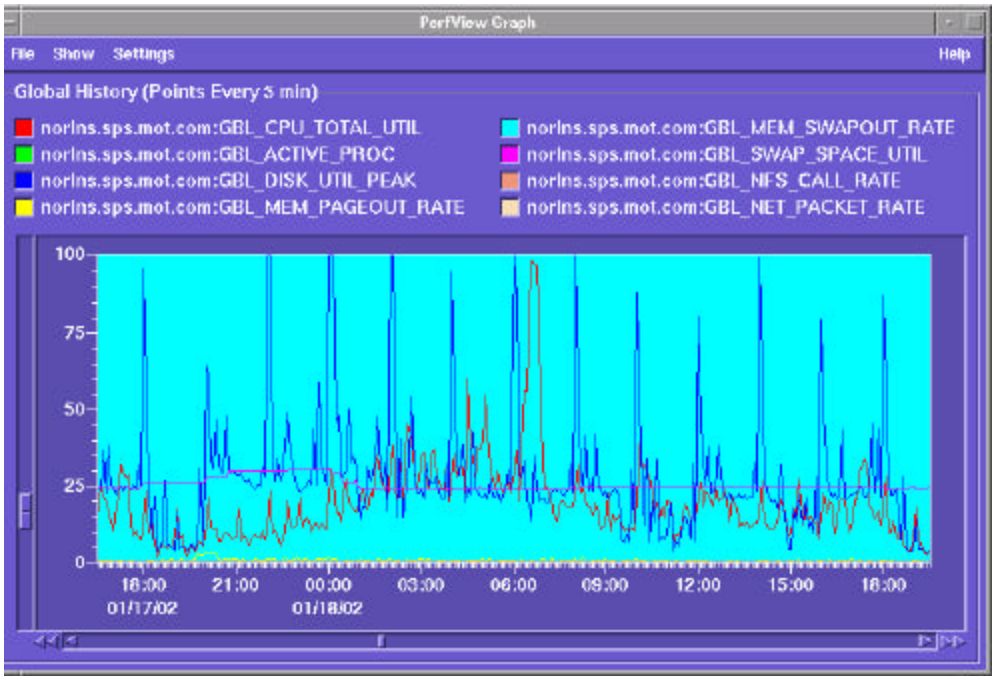
In case of performance difficulties, a high queue depth can indicate the general area of an issue. As an example, if the global disk subsystem queue is high, there may be an issue with the SAN, with the array, or with components in between. If one of the other queues is high, while the disk queue remains low, the performance problem is most likely not on the disk subsystem.

PerfView Analyzer is a very powerful tool to gauge the effectiveness of a re-allocation. As an example, consider the following Global History

Figure 4 – Global Depth Chart

chart taken one month before re-allocating storage:

This Before Migration Global Summary (Figure 5) depicts a much more granular view of the system's performance. The CPU is seen to run at



approximately 15 – 25 %, and disk utilization runs 25%, peaking

Figure 5 – Premigration Global History

frequently at 100%. This data was fairly representative of day-by-day operations of the system over several months. Before our re-allocation operation, this system utilized disks that were both striped and unstriped, and fairly similar to our earlier sar example with "system1".

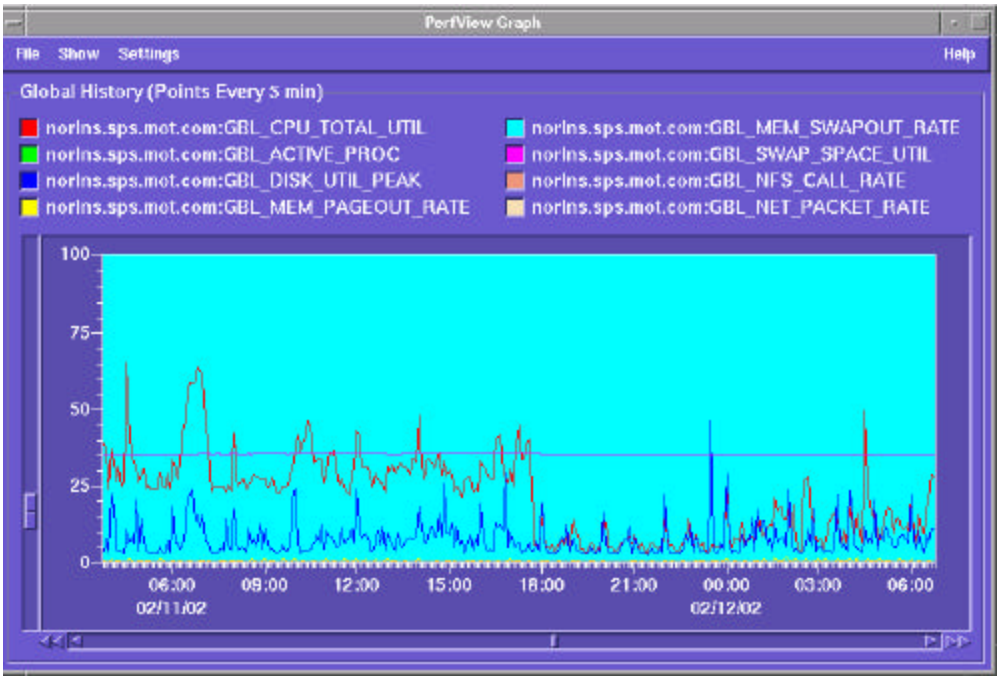
We evaluated this system using the techniques presented earlier in this paper, and, using the sar analysis, showed an improvement in disk performance. PerfView allowed us an opportunity to present an amazing impact on the overall performance of the system as a result of our re-allocation in the After Migration Global Summary (Figure 6).

Points to note:

- Figure 6 depicts a point at which the obsolete disks have been removed, and the re-mirroring operation is fully completed to the

new allocation. The drop in CPU utilization at 18:00 is the end of this operation.

Figure 6 – Post Migration Global History



- Disk Utilization has dropped significantly – averaging 5%, peaking at 45%
- CPU Utilization has dropped significantly – dropping to below 15% on average.

Summary

Using sound storage and systems management techniques, we built processes to effectively increase storage utilization efficiency. By utilizing a design phase, we carefully mapped out all storage requirements, and developed an implementation plan to take advantage of the storage architecture. Implementation of the plan can be conducted using various techniques based on operational conditions or factors. After completion, performance tools can be used to validate and meter the effectiveness of the strategy.